Uncontrollable: A User Experience Design Proposal for a Hands-Free Gaming Accessibility Framework

by

Andrea Brucculeri

Department of Art, Art History and Visual Studies
Duke University

Date:_______________________
Approved:

____
Victoria Szabo, Advisor

____
Augustus Wendell

____
Mark Olson

Thesis submitted in partial fulfillment of
the requirements for the degree of Master of Arts in the Department of
Art, Art History and Visual Studies in the Graduate School
of Duke University

2021

ABSTRACT

Uncontrollable: A User Experience Design Proposal for a Hands-Free Gaming Accessibility Framework

by

Andrea Brucculeri

Department of Art, Art History and Visual Studies
Duke University

Date:_______________________
Approved:

____
Victoria Szabo, Advisor

____
Augustus Wendell

____
Mark Olson

An abstract of a thesis submitted in partial
fulfillment of the requirements for the degree
of Master of Arts in the Department of
Art, Art History and Visual Studies in the Graduate School of
Duke University

2021

Abstract

The technology necessary for hands-free video gaming is available, especially for head-gesture-centric controls. However, remapping controls to head gestures ranges from frustratingly tedious to impossible. I propose a common language of gestures and game actions that categorizes the controls by their use frequency. My suggested categories are primary, secondary, tertiary, and quaternary. The most preferred gestures and more frequently used controls are classified as primary while the least preferred gestures and least frequently used controls classified as quaternary. I also propose constructing an interface software and API to gather data from game designers, hardware designers, and users to suggest optimized game controls for users requiring accessibility. I created a demo for one branch of this for my technical project -- a game to help players determine which gestures they can perform the most accurately so that these gestures may be paired with controls most vital to successful gameplay.

Dedication

This thesis is dedicated to my mom and dad, who taught me that love really is the most important ingredient.

Contents

List of Tables

List of Figures

Acknowledgements

While the events of 2020 separated us physically, my thesis was not written in a vacuum. I have many people to thank for their contributions, both big and small. Just a few are listed here.

Victoria Szabo for being a confident leader and source of inspiration and reassurance while guiding me through the master's program.

Augustus Wendell for always pushing me to learn more through his remarkable and eye-opening teachings, both technical and theoretical.

Mark Olson for being a trusted teacher, advisor, and friend, who has made a much greater impact on my life than Cheerios.

Bill Seaman for his outstanding help with researching hands-free control systems and his brilliant imagination for future technologies.

Hannah Jacobs for being a wise and trusted mentor while encouraging me to get more involved with the Duke community and the Wired Lab.

Ryan Armstrong for his dauntless support and for being a reliable, compassionate listener and teammate.

Madeleine Nutt for bringing me laughter and unshakeable friendship while learning and growing by my side.

Gary Kayye, Scott Geier, and countless other professors at UNC-Chapel Hill for giving me the passion and empowerment to tackle seemingly impossible problems.

1. Background

Over 3 billion people play video games.[1] That number captures people of all ages playing on consoles, computers and smartphones from all over the world, and the industry is showing no signs of slowing down. An ESA survey estimates that 214 million Americans play video games, including 65% of adults, with one reason being that they help foster familial bonds.[2] Now more than ever, video games are advertised as entertainment that the whole family can join in on -- everyone from a small child who cannot yet speak to an older adult with limited video game experience. Gaming companies who once targeted young boys now encourage all audiences to partake in the virtual adventures, and women and minorities have seen greater representation in games over the decades. However, video games have long eluded those who cannot or do not use their hands, as virtually all video game controllers (whether they be keyboards, cell phones or traditional controllers) require the use of hands. There are some alternatives including voice controllers, sip and puff sensors, and various methods that respond to facial movements. Nonetheless, these methods can be difficult to acquire

[1] "Global Video Game Consumer Population Passes 3 Billion," *DFC Intelligence*, 14 August 2020.

[2] Stanley Pierre-Louis, "ESA Leadership Desk: 214 Million Americans Play Video Games," *Entertainment Software Association*, 2020.

and pose their own challenges. The wide-spread use of games only serves to further alienate people who cannot participate.

1.1 Why games?

Indeed, why games— especially video games, which have been criticized for promoting violence, alienation, and illiteracy, among other issues. This thesis does not assert that there are no negative consequences to video games' prominence in culture, especially when they are played extremely often. Like any activity, video games should be played with healthy moderation and not all types of games are best for all types of people. However, this thesis is not about telling people what they should or should not do with their spare time. My research is about exploring ways to make the common hobby of video games more accessible to those who would like to participate.

For those who do not understand the appeal of video games and why people with disabilities are willing to work against challenges to play them, I will briefly explain the positive psychological impacts of gaming. Video games have proven to be more useful than simple fun— games help people learn, relax, and bond with others. Games can even empower and inspire people. They give people great surges of joy and thrill as well as feelings of accomplishment and pride, all without requiring the user to leave their couch. This is why being left out of gaming can feel so alienating – games offer experiences that are difficult to achieve in similarly low-movement situations.

Video games have fostered tight communities and been a source of strength and family to individuals feeling isolated. During difficult times, games can offer escape, solace, and meaning along with entertainment. One of the reasons people can be so emotionally invested in gaming is that the core principles of a game are similar to the core requirements for people to feel the state of joyful productivity known as “flow.”

1.1.1 Games and Flow

First I will describe the basic principles of games, then how these principles relate to flow. There are a few definitions of games that share key points (this refers to games in general, not just video games). Veteran game developer Katie Salen in her book *Rules of Play: Game Design Fundamentals* says, “A game is a system in which players engage in an artificial conflict, defined by rules, that results in a quantifiable outcome.”[3] Game designer Jane McGonigal says that there are four qualifications for a game: a goal players will work to achieve, rules providing limitations, a feedback system giving player progress, and voluntary participation.[4] Play theorist Brian Sutton-Smith says, “at its most elementary level then we can define ‘game’ as an exercise of voluntary control systems in which there is an opposition between forces, confined by a procedure and rules in order to produce a disequilibrial outcome.”[5]

[3] Katie Salen, *Rules of Play: Game Design Fundamentals*, p. 80.

[4] Jane McGonigal, *Reality is Broken*, Chapter 1.

[5] Avedon, Elliott M., and Brian Sutton-Smith, *The Study of Games*. p. 405.

Based on these definitions, a game must include a clear goal, rules that provide limitations, measurable feedback, and voluntary participation. The qualifications for something to be a game are similar to the qualifications for flow.

Flow is a state coined by Mihaly Csikszentmihalyi in his book *Flow: The Psychology of Optimal Experience*. He defines flow as "a state in which people are so involved in an activity that nothing else seems to matter; the experience is so enjoyable that people will continue to do it even at great cost, for the sheer sake of doing it."[6]

There are four main qualifications for someone to enter a state of flow. The first is clear goals, or knowing exactly what is expected to be accomplished. The second is frequent feedback. This allows a person can know when they are doing well or poorly and adjust accordingly. The third is low stakes for failure. Fear of failure reduces willingness to take risks, impedes learning, and dampens creativity. The fourth (and perhaps most important) is a balance between challenge and skills. People in a state of flow are challenged enough to feel productive and proud of the work they are accomplishing, but not so challenged that they experience prolonged frustration and hopelessness.

[6] Mihaly Cskikszentmihalyi, *Flow: The Psychology of Optimal Experience*. 1990, p.4.

Flow is desirable in schools, jobs, and games alike. People in a state of flow are highly motivated, cooperative, productive, excited, and in an optimal state to learn and grow. Above all, flow feels good – people want to feel productive and enjoy their work. However flow can be elusive, especially for people struggling to find a fulfilling occupation. This is the case for many people with disabilities, as the employment-population ratio for people with disabilities is 19.3% as of 2019. This is in strong contrast to the employment-population ratio for people without disabilities, which is 66.3% for 2019.[7] Video games can help people experience flow when they cannot get it elsewhere, and games offer feelings of enrichment and success with minimal financial and physical commitment.

1.1.2 Fiero

Flow helps people feel joyfully productive, but is one more important emotional state that contributes to happiness in working and learning: fiero.

Fiero is an Italian word that directly translates to "pride," but its meaning is more powerful than pride. McGonigal describes fiero as "what we feel after we triumph over adversity. You know it when you feel it — and when you see it. That's because we almost all express fiero in exactly the same way: we throw our arms over our head and

[7] Bureau of Labor Statistics. *Persons with a Disability: Labor Force Characteristics -- 2019*, pages 1–2.

yell."[8] This is the emotion people chase while working on a task, competing, or generally trying to accomplish something. Fiero can be as physically dramatic as scoring a touchdown or as simple as finally getting the cookie recipe just right. Both of these events stir up the same rush of emotion – the feeling of success. Having fiero means having succeeded.

Video games (and many tabletop games) are extremely good at providing people with fiero. This is because fiero is the direct result of succeeding at a goal, and video games are excellent at providing a multitude of reachable goals, from small "daily" goals to long-term projects that require dozens of hours. Fiero is also credited with being the emotion that "keeps people coming back for more," so to speak. The lack thereof causes people to lose interest in an activity because, naturally, no one wants to work on a task when they have no hope of success. People love fiero, but are often completely deprived of it – their goals at work or in life are so massive and long-term that they are unreachable or cannot provide the person with a sense of progress, hope, or meaning. Video games provide people with an opportunity to experience fiero without making major or impossible life changes, such as switching jobs or altering the circumstances of their life.

[8] Jane McGonigal, *Reality is Broken*, page 33.

1.1.3 Games and the Secrets to Happiness

So it seems video games can help people feel productive, excited, and proud – but can they actually make people happy? The source of happiness is an infinitely massive topic that will in no way be covered in this thesis. To explore the link between games and happiness, I will again refer to game designer Jane McGonigal who explains what she calls the "four secrets to making our own happiness."[9]

The first secret is daily satisfying work, which differs between people, but always means "being immersed in clearly defined, demanding activities" that allow us to see the direct impact of our effort. Games are usually fashioned such that players are immediately aware of their impact and the consequences of their actions. This is the "feedback" requirement discussed in the previous section. The second secret is the experience or ongoing hope of being successful. McGonigal describes this as us wanting "to feel powerful in our own lives" and "be optimistic about our own chances for success, to aspire to something, and to feel like we're getting better over time." Consider this in relation to the "balance between challenge and skills" requirement for flow—people experiencing such balance have constantly renewed hope in their ability to succeed and do occasionally reach success (sparking fiero). The third secret is social connections. Even the most introverted among us still wish "to share experiences and

[9] Jane McGonigal, *Reality is Broken*, page 45-51.

build bonds," as this is where most of our happiness comes from. Video games are not often thought of as social, but many games have thriving online communities with tight bonds. Some games require people to play with friends or strangers to accomplish tasks, and others simply simulate socialization by allowing the player to make friends with virtual characters. The fourth secret is meaning, or "the chance to be a part of something greater than ourselves." This is from the natural craving to belong and contribute to a community. As mentioned before, the world of video games is a huge, enthusiastic community composed of thousands of sub-communities playing specific games or genres. These communities are part of what can turn a one-time gamer into a life-long gamer.

Table 1: Summary of Games, Secrets to Happiness, Flow and Fiero Information

Requirements to be a Game	*Requirements for Flow / Fiero Loop*	*McGonigal's Secrets to Happiness*
Clear goals	Clear, reachable goals	Daily satisfying work
Measurable feedback	Frequent feedback	Meaning
Rules providing limitations	A balance between challenge and skills	Experience or ongoing hope of success
Voluntary participation	Low stakes for failure	Social connections

Games can increase happiness by helping us feel productive, meaningful, successful and social in our virtual worlds. Even if the productivity is artificial, it is enough to cause delight and satisfaction, especially in people who do not feel that they are otherwise participating in productive, meaningful work. People with disabilities may be particularly susceptible to these feelings as they are at an increased risk for depression.

A 2017 study found that the prevalence of depression in adults without disabilities is between 22.8 and 27.5 percent, while the prevalence in adults with disabilities is estimated from 24.9 to a staggering 41 percent. Further, individuals with physical disabilities reported having more pain, anxiety, depression, and a lower quality of life than those without.[10] Video games are not the answer to all of these problems, but games could be a source of joy and empowerment for those with disabilities struggling with their mental health. In an article by *The Washington Post* titled, "'It's my escape.' How video games help people cope with disabilities," journalist Hawken Miller writes that, "gaming allows many [people with disabilities] to do things in a virtual space they could only dream of in reality. It also helps them connect and overcome social anxiety and feelings of depression." The following excerpt is from this article:

[10] Shen, Szu-Ching. "Incidence, Risk, and Associated Factors of Depression in Adults with Physical and Sensory Disabilities: A Nationwide Population-Based Study." 2017.

> "It's my escape," said Brian "Wheely" McDonald, 31, who has arthrogryposis, causing the normally elastic tendons in his hands to stiffen. "I'm not disabled in video games. I have people telling me all the time how amazing I am at games."
>
> Mark Barlet, founder of Able Gamers Charity, which helps those with disabilities connect to video games through adaptive technology, said that like with many able-bodied individuals, video games can help those with disabilities forget what they can't do.
>
> "The trappings, the scary parts of disability, don't define and don't have to define you [in video games], and that's really breathtaking," Barlet said.
>
> It's proven that social interaction, now readily available through online connectivity in video games, leads to better health outcomes. Researchers at the Health Resources and Services Administration compared social isolation to smoking 15 cigarettes every day.[11]

Brian McDonald is far from the only gamer sharing stories about how video games have helped him cope with disabilities. There are thousands similar stories to be found in online forms and articles. However, if video games were accessible enough for every person with disabilities to play, this thesis would not exist. The following 100 or so pages will describe the current state of video game inclusivity, gaming technology, hands-free gaming options, controller designs, my suggestion for a game controller design solution, and much more. With this brief understanding of the importance of video game accessibility, let's move into the complex and boundless world of improving it.

[11] Miller, Hawken. "'It's My Escape.' How Video Games Help People Cope with Disabilities." *The Washington Post*, 10 Dec. 2019.

1.2 Thesis Statement

As gaming becomes more popular and more relevant to a person's social and cultural involvement, the pressure to makes games more accessible is growing faster than the software necessary to increase accessibility. Hands-free gaming accessibility, an area lacking attention thanks to the common use of traditional hand controllers, could be the most easily achieved by creating universality in the way people talk about and design game controls. This thesis creates a framework on how to design such a game control experience.

1.3 Summary of Arguments

Video games were initially very exclusionary, designed for young white boys from upper-middle-class families. Over time, game developers have made more and more efforts to include women and minorities in games and gaming communities. Nonetheless, there is still a long way to go. This great inclusivity wave for video games has made gamers with disabilities ever more excluded. This is especially true for gamers who cannot use their hands to play.

Virtual reality is both especially exclusive toward people with disabilities (as people must be one with the virtual body) and potentially the answer to hands-free gaming. By advertising and design, current virtual reality headsets, interface designs and games highly favor abled people. However, VR allows people to communicate with the machine using head movements, and many headsets are already equipped with other useful sensors such as eye-trackers and microphones. These sensors suggest that virtual reality headsets have great potential for hands-free gaming.

Many different gestures could replace hand gestures for gaming purposes. Breaking down games by primary, secondary, tertiary and quaternary actions according to how frequently they are used helps people determine which gestures may be better for which buttons on any particular game. This is because each person is an individual and will have their own unique range of comfortable abilities, and each game will have a unique set of buttons that are used more frequently than others.

Likewise, a person must become familiar with what gestures work best for their gaming purposes. It would be useful to have a game to help determine this. Such is the foundation of my technical project -- a diagnostic game that could use face-tracking, speech, eye-tracking, etc. to measure the dexterity of a person's facial gestures and recommend which of their gestures would be best for primary controls, secondary controls, etc. My technical project is a part of a larger interface software concept I have envisioned. This software would harness data from game designers, hardware, and gamers to produce individually-optimized control suggestions for each game for a particular gamer.

The future of gaming accessibility will be shaped by today's young designers, engineers and gamers. Current solutions lack in adaptability and versatility. My project contributes directly to the gaming industry by challenging game control normativity, refocusing game controls on universal design, and encouraging game design that has the success and enjoyment of the individual user at heart, no matter their abilities.

Games will never become as accessible as they could be without a partnership between game designers, hardware designers, and gamers with disabilities. The technology to create great adaptive gaming already exists; we just haven't brought all the required resources together to make great accessible software. Hands-free gaming will only become more important as time goes on because as this gaming generation ages, more people will become disabled and require new equipment to game. Technology such as I have described will be vital to keeping gamers connected to their communities into old age.

2. Video Game Progress and Exclusion

The shift toward a more inclusive gaming world started in the early 2000s. Noah Wardrip-Fruin, a professor of Computational Media at the University of California Santa Cruz, wrote in 2004, "computer games are themselves in a period of considerable development and redefinition… the channels of game interaction are multiplying, and as new artistic and marking methods arise and the first generation of computer gamers come of age, games are reaching and being developed for an ever-broadening demographic."[1] As the decades have passed, great strides have been made toward a gaming space that is welcoming to all, but the gaming community is still struggling to leave behind its history of racism and sexism.

2.1 Gender Diversity in Gaming

Male-dominated video games are nothing new, especially for large companies such as Nintendo. Jesse Divnich, the vice president of insights and analysis for Electronic Entertainment Design and Research, said "The Nintendo Entertainment System was targeted toward boys under 10. If you look at the Super NES five years later, it starts targeting boys ages 10-15."[2] The Nintendo Entertainment System (NES) was one of Nintendo's first gaming systems. The Super NES was released in 1990 alongside a console named the Gameboy, solidifying Nintendo's stance on its target audience. At the

[1] Noah Wardrip-Fruin, *First Person,* p. 4.
[2] Tracy Lien, "No Girls Allowed," *Polygon*, 2 December 2013.

time, Nintendo ran TV ads targeted at young boys by featuring boys playing games with masculine characters. As the boys who played games grew older, ads and consoles were targeted toward older boys, aiming to make them life-long players. However, girls and feminine characters rarely (if ever) made an appearance in games or advertisements.

Games and culture researcher Amanda Cote offers an explanation for how Nintendo might have contributed to how "gamers" became decidedly male-only. "A detailed content analysis of Nintendo Power issues published from 1994-1999 shows that mainstream companies largely ignored the girls' games movement, instead targeting male audiences through player representations, sexualized female characters, magazine covers featuring men, and predominantly male authors," she wrote.[3] This lack of representation in the press ultimately affected a girl's ability to identify as a gamer, as this was never seen within the gaming community.

Men and boys also dominate gaming competitions. At a *Street Fighter* live-streamed gaming competition, Miranda Pakozdi (one of only two female competitors) was increasingly heckled and harassed during gameplay by competitive fighting gamer Aris Bakhtanians.[4] Aris asked her why she wasn't wearing a skirt, offering to buy her one so he could see her in it, and urged her to take her shirt off. She was called a distraction for having "girl parts." Aris leaned up against her, smelling her and

[3] Amanda Cote, "Writing 'Gamers': The Gendered Construction of Gamer Identity in Nintendo Power," p. 479.

[4] *GTFO: Get The Fuck Out*, 2015.

commenting on her scent. He asked her to say her bra size, and then demanded that she learn how to play the game while being continuously harassed. She ended up throwing her match and tweeting that she would have left immediately if she were not contractually obligated to stay. Later, Aris was asked in an interview, “Can I get my *Street Fighter* without sexual harassment?” He responded, “You can’t. You can’t because they are one and the same thing. This is a community that’s, you know, 15 or 20 years old, and the sexual harassment is part of a culture.”

Unfortunately, Miranda’s experience is not unique or near as extreme as sexual harassment can be in gaming culture. Her story and dozens of others are covered in the 2015 gender in gaming documentary *GTFO: Get The Fuck Out*. Todd Harper, author of *The Culture of Digital Fighting Games,* explains in the documentary that the root of the harassment problem is that, for decades, video games were a boys-only space. Men are making the games, testing the games, playing the games and selling the games. “It’s a self-perpetuating cycle,” he says. “Who’s buying games? Straight white men in their 20s. So we’re going to make a game targeted toward straight white men in their 20s. Amazingly, that means more straight white men in their 20s are going to buy your game.”[5]

[5] *GTFO: Get The Fuck Out*, 2015.

To stop the harassment, the cycle needs to be broken. Today, 50% of people who play games are female, yet only 21% of game developers are women.[6] Targeting girls and women in advertising will not be enough. In her book *Gaming Representation: Race Gender and Sexuality in Video Games*, Jennifer Malkowski explains that there is no one fix to females being "aggressively disrespected" in multiplayer gaming, but that doesn't mean there is no hope. "Obviously, creating games that feature femmes fatales will not dismantle this formidable web of gender trouble in the video game industry," Malkowski wrote, referring to the women with "smarts, sex appeal, ambition," and other admirable traits who are so often and easily portrayed in contemporary cinema. "But many who are working to alleviate these problems do advocate for increasing the number of nuanced, powerful, and playable female characters in games as one core strategy."[7] One of the points she continues to touch on is the problem of sexualizing the few female characters that are in video games. It seems women and sex are always paired in video game contexts. In his article "Keeping Abreast Hypersexuality: A Video Game Character Content Analysis," Edward Downs conducted a character gender analysis. He found that "In comparison to male characters, females were significantly more likely to be shown partially nude, featured with an unrealistic body image, and depicted wearing sexually revealing clothing and inappropriate attire."[8] For the scope of

[6] International Game Developers Association 2017 Developer Satisfaction Survey.

[7] Jennifer Malkowski, "Gaming Representation: Race Gender and Sexuality in Video Games," p. 20.

[8] Edward Downs, "Keeping Abreast Hypersexuality: A Video Game Character Content Analysis," p. 721.

this paper, I cannot cover this topic entirely but nonetheless found it important to note that the appearance of the female characters is contributing to the problem.

Noah Wardrip-Fruin writes about why games may not be taken as mere games. Games simulate reality, and that can perpetuate problems intentionally or unintentionally as he says that "all simulations are political."[9] *Super Smash Bros*. is a game that simulates a rough-and-tough competition. By creating a space void of female characters, the game perpetuates the idea that women are not tough enough or competitive enough to participate. Peddling this idea in an already male-dominated space does no favors for women and girls, who already face an upward battle to prove their worthiness as players.

Fortunately, things have changed in the gaming universe since the Nintendo Entertainment System was released in 1985. Nintendo released their newest console in 2016 — the Nintendo Switch. Advertisements for the Switch feature boys and girls of various ages and races playing several games, making for a much broader target audience than early Nintendo ads. *Super Smash Bros. Ultimate*, released for the Switch in 2018 as the latest edition of the game series, also has advertisements showing significant player diversity. From the first game release in the *Super Smash Bros.* series in 1999 to the newest game release in 2018, there was a 22.6% decrease in playable male characters,

[9] Noah Wardrip-Fruin, *Expressive Processing: Digital Fictions, Computer Games, and Software Studies*, p. 303.

making room for more female, non-binary and androgenous or gender non-conforming characters (See Appendix B). This increase in awareness from Nintendo is echoing throughout the industry, showing promise for more gender equality in the future of gaming.

2.2 Racial Diversity in Gaming

Like other media industries, the video game industry has had racial representation problems since its inception. As gaming and African American culture ressearcher Treandrea Russworm writes, this is both a quantitative and qualitative problem. While the number of black avatars in video games may be increasing, the narratives surrounding them are not changing quickly. Russworm explains that "the representation problem has less to do with the pitiably limited number of characters of color who appear as playable protagonists in games and more to do with the ideological work those characters facilitate, including how gamers feel about and express empathy for such characters when they do happen to appear in games across genres."[10] This is seen clearly in games about sports. These games appear diverse because they often feature a high percentage of black people, most of which are modeled after real-life professional athletes, such as LeBron James and Michael Jordan. David Leonard, Professor of Comparative Ethnic Studies at Washington State University, explains that

[10] Treandrea Russworm, "NBA 2K16: Race."

"eight out of ten black male video game characters are sports competitors," suggesting that the number of black video game characters is not a meaningful way to track racial representation in video games overall.[11] The quality and nature of the representation are highly consequential to the audience's perception.

If the representation promotes harmful stereotypes or confines avatars of color to specific characteristics and environments, then the representation only impairs the progress toward a more inclusive gaming community. This was a major problem in the early 2000s and earlier, before gaming companies started expanding their target audiences. A 2002 study concluded that "African-American characters are more likely to display aggressive behaviors in sports games (i.e., trash-talking and pushing) than whites. More than 90% of African-American women function as props, bystanders, or victims."[12] Leonard points out how the 1999 boxing game *Ready to Rumble* includes characters of a variety of races, but only in poor, stereotypical depictions. He writes that the characters include, "Afro Thunder, a gigantic Afro-wearing boxer, who is more adept at talking trash than fighting… a Hawaiian sumo wrestler who speaks poor English and has slanty eyes, a heavy accented Croatian immigrant, and a Mexican boxer named Angel (raging) Rivera."[13]

[11] David Leonard, "Performing Blackness: Virtual Sports and Becoming the Other in an Era of White Supremacy," p. 321, 322.

[12] Children Now, "Fair play? Violence, gender and race in video games."

[13] David Leonard, "Live in your world, play in ours": Race, video games, and consuming the other," Nov 2003.

While there are still stereotypes plaguing minorities in video games, some progress has been made over the past decade. Apex Legends, a game with over 70 million registered users as of October 2019, features mixed race, black, Asian, gay and non-binary characters with only three white characters. There are two black female characters celebrated for their combat and medic skills, ambitious attitudes, and nuanced personalities. Fortnite, a third-person shooter game with over 350 million players worldwide, randomly assigns skins (or the appearance of the avatar) to the player with each match unless the player pays for a specific avatar. These random skins include black, Hispanic, and mixed-race characters, all of which are equal in skill and utility during play. Another way some games have addressed racial representation is by allowing more avatar customization. Players can pick from a number of skin tones, hairstyles, and facial features to create a unique character. This is seen in casual games *Stardew Valley* and *Animal Crossing: New Horizons* as well as the popular first-person adventure *Skyrim* and the latest installment to the Pokémon franchise, *Pokémon Sword and Shield*. These are just a few of the many games that allow avatar customizations.

Fixing racial representation in video games will have to start outside the games – in the communities that the players live in. Protests across the United States in the summer of 2020 inspired many game companies, including Electronic Arts, Epic Games and Sony Interactive Entertainment/PlayStation, to declare their support for people of color and make donations to social justice organizations. These actions are helpful but will not be enough to fix the racial representation problems facing video games today. The International Game Developers Association found in a 2019 survey that, among game developers worldwide, 81% identify as white/Caucasian/European, 7% identify as Hispanic/Latinx, and only 2% identify as Black/African-American/African/Afro-Caribbean.[14] Better representation in games will require better representation in game companies as well. As video game companies become more focused on creating inclusive spaces, we will expect – and demand – to see better racial representation in the years to come.

2.3 Physical Accessibility in Video Games

Gamers have been witnessing the slow but determined march toward gender, racial, and sexuality inclusivity for years. These strides have changed the industry by impacting both the characters in the games and the people working behind the scenes. Nowadays, games are advertised for "everyone" -- all ages, all genders, and all cultures worldwide. But in this "games for everyone" frenzy, gamers with disabilities have been evermore isolated and excluded. Perhaps this was less apparent when video games were only intended for white boys under 12 years old, but now the involvement of so many people in games just brings more attention to how gamers with disabilities struggle to be a part of the community the way others can. There are over 33 million gamers with

[14] The International Game Developers Association, Developer Satisfaction Survey 2019 Summary Report.

disabilities in the United States alone.[15] While people with disabilities are also seriously underrepresented in video games, it is difficult to start discussing this when so many people with disabilities physically cannot play the games to begin with. In *Mismatch: How Inclusion Shapes Design,* author Kat Holmes talks to John Porter, a gamer with motion disabilities, about inclusion in video games. The following is Porter talking about his experience playing video games with limited physical mobility:

> "All of these games are based on the broad assumption that you'll be using your fingers and your hands to interact. And for me, that is almost entirely moot… Whenever I'm using technology, I use speech to control it even though it was never designed to work for speech. It's not just that a game isn't optimized for my abilities. It was made without ever even considering the possibility that someone would need to interact with it in the ways that I do. That places the onus on me to figure out all these workarounds. For gamers with disabilities, we have to spend as much time figuring out how to play a game as we do actually playing it."[16]

This is part of my inspiration for this project – suggesting a design concept that helps players with disabilities spend less time trying to figure out workarounds and more time enjoying their game. One way to see some of the ideas and home-made inventions is to scroll through posts on the r/disabledgamers page of Reddit.[17] If you do this, the first thing you may notice is that the large number of posts from people seeking help or advice with hands-free gaming options. Scrolling through this subreddit as I write this, I see posts titled, "Hands-Free Hollow Knight," "Hand-amputees, what is

[15] AbleGamers Charity. "AbleGamers and Gamers Outreach Partner to Help Children with Disabilities in Need," 2017.
[16] Kat Holmes, *Mismatch: How Inclusion Shapes Design*, p 49.
[17] Link to the disabled gamers subreddit: https://www.reddit.com/r/disabledgamers/

your solution to using VR controllers," "PS4 with tremors," and a man describing that his wife has no function in her right hand due to a stroke and is looking for alternative controllers. Each person describes their own reason for wanting a hands-free device, but their problems are the same – the available controllers and hands-free gaming options just aren't adaptable enough for them. Modern controllers are too complex to find easy adaptations, and every gamer must research unique solutions to each gaming system and game.

Video games have also become more complex in design over time. Early games, such as *Pong* and *Pac-Man*, relied on very few controls as they were based mainly on using one four-direction stick. This is relatively accessible compared to video game controllers produced today, such as the Nintendo Switch Pro controller or Xbox One controller. These products have over a dozen buttons and two analog sticks, largely requiring two dexterous hands to operate. While someone with a motion disability may still find *Pong* perfectly easy to play using one large stick, the same person could have difficulty using modern controllers to play more complex games.

Some companies have taken steps to increase accessibility by making less complex versions of their current controllers, although this does not benefit people who cannot use their hands. Microsoft launched the Xbox adaptive controller in 2018, a device that allows full customization of its buttons, including sensitivity controls. A commercial for the controller aired during Super Bowl LIII featuring disabled children playing video games using the device.[18] Microsoft received praise from thousands of people (including some celebrities) across social media platforms, but not everyone was so impressed. Josh Straub, creator of the physical accessibility review website Dagers[19], told *The Washington Post*, "In the area of hardware, the [Xbox adaptive controller] is a great device, but it only helps one segment of the disabled gamer population. And yet the narrative seems to be that now everyone can play, because Microsoft released this device."[20]

This narrative – that one device or one change can make gaming accessible for everyone – is part of the struggles facing gamers with disabilities. It is important to understand that there is no single way to make games, or anything else, 100% accessible. The path to more accessibility will be long, slow, expensive, and never enough to please everyone. This is not to be pessimistic but to ground expectations in reality.

Holmes tells a story about the United States Air Force (USFA) that illuminates the trouble with trying to create one solution that works for everyone.

> In the 1940s, the first fighter jets were designed to fit the average pilot. The USFA measured hundreds of bodily dimensions across thousands of pilots and used the averages of that data to design the instruments of the flight deck, or cockpit.
>
> Every feature of that flight deck was fixed in place, without adjustability. The assumption was that any individual pilot could just adjust himself to overcome

[18] A link to the commercial, posted on iSpot.tv: https://www.ispot.tv/ad/IS6b/microsoft-xbox-adaptive-controller-super-bowl-2019-we-all-win

[19] Dagers website link: https://dagersystem.com/

[20] Stoner, Grant. "How Accessibility Consultants Are Building a More Inclusive Video Game Industry behind the Scenes." *The Washington Post.*

> the gap in reaching any element of the flight deck that wasn’t a perfect fit for him.
>
> However, the Air Force was experiencing a high rate of crashes that couldn’t be attributed to mechanical failure or pilot error. A Lieutenant and researcher, Gilbert Daniels, studied just ten of those human dimensions that were used in the design of the original flight deck. He measured four thousand pilots to confirm how many of them fit all ten dimensions.
>
> The answer was zero. Not a single pilot matched all ten dimensions.[21]

This eventually led to new adaptable designs that each pilot could adjust to fit their own body, including seat belts, seat heights, and the position of controls. These changes led to improvements in both performance and safety.

A similar approach needs to be taken with video games. To make the flight deck safer and easier to use, many adjustments had to be made and parts changed to become adaptable. As a result, flying is now a better experience for all pilots. With the right changes, video games can become better for everyone. But we will have to make changes to more than just controllers – we will need to see changes throughout the entire game design process.

[21] Kat Holmes, *Mismatch: How Inclusion Shapes Design*, p. 97.

3. Virtual Reality's Accessibility Dichotomy

One of the great affordances of virtual reality technology is that the headsets' motion sensors allow the user to communicate with the machine without using their hands. Yet even the newest virtual reality games and headsets heavily rely on hand controllers for navigation and gameplay. The image below (from the company website) is of the 2020 Vive Cosmos, a cutting-edge piece of VR technology equipped with "six camera sensors and the latest software optimizations to enhance inside-out tracking accuracy," yet it still requires the use of two hand controllers.[1] Virtual reality headset advertising even targets people without disabilities by running ads of people leaping through virtual battlefields and ducking to doge enemy projectiles. Controlling the virtual body using intuitive physical movements is a noteworthy technical feat – but not for those who relied on a lack of physical movement for comfortable gaming. In this way, virtual reality presents a tense dichotomy between becoming a highly-accessible technology for hands-free gaming, and the most alienating yet.

[1] Image and quote of Vive Cosmos from the company website: https://www.vive.com/us/product/vive-cosmos/features/

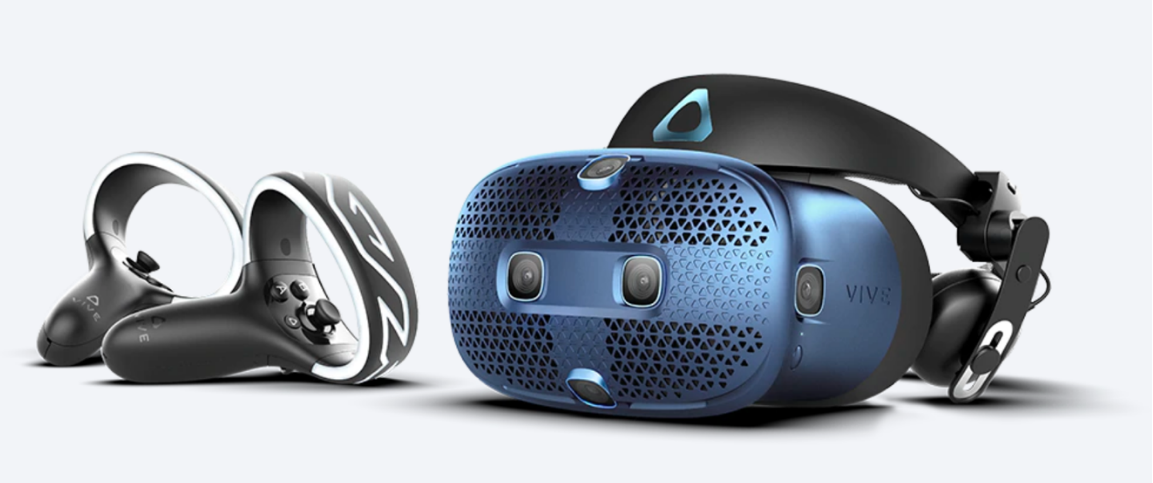


Figure 1: Vive Cosmos VR headset and controllers, 2020

VR technology is improving rapidly, and as more sensors are added to the systems and current sensors improve, new possibilities for hands-free gaming arise. These additions and improvements will not only make gaming more accessible, but a more comfortable and enjoyable experience for everyone. Jeremy Bailenson, founding director of Stanford University's Virtual Human Interaction Lab, wrote, "VR will not only give us access to experiences that are difficult to obtain, it will also allow us to see impossible things, fantastic things, things that will allow us to see the real world in new ways and allow us to stretch out minds beyond anything we can imagine." [2]

3.1 Advertising Virtual Reality

Virtual reality headsets are equipped with a multitude of sensors, including cameras, microphones, gyroscopes, accelerometers, magnetometers, eye-trackers,

[2] Jeremy Bailenson, *Experience on Demand,* p. 87.

structured light systems and more. Yet virtual reality games heavily rely on hand controllers for navigation and gameplay, stripping the technology of its natural potential for hands-free use. Part of the reason hand controllers are required for these games is that the gameplay requires the user to use virtual hands to accomplish tasks. These high-movement games are heavily advertised for their flashy action-packed scenes, further solidifying the ty between virtual reality and inaccessibility.

Virtual reality headset advertising targets people without disabilities by running ads that emphasize full-body movement in dramatic environments. One popular game called *Beat Saber* appears in numerous VR commercials, showing players ducking, jumping, and swinging their arms across space as they slice open colorful cubes. Other commercials depict people playing virtual sports such as tennis and bowling, or involved in dramatic fight scenes as they dodge bullets and attack enemies.

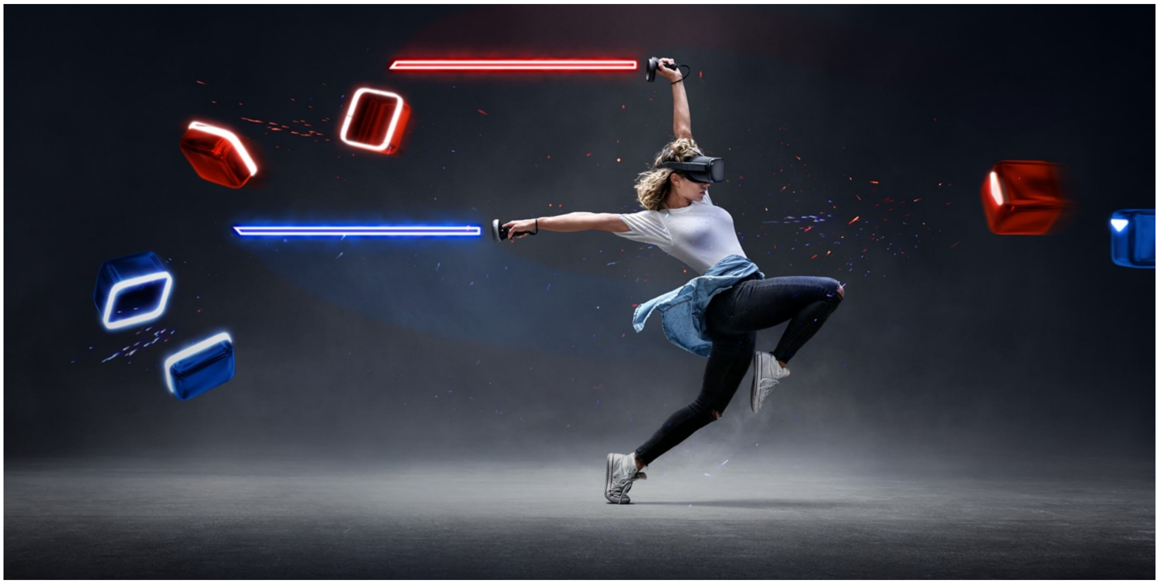


Figure 2: Oculus Quest VR ad for *Beat Saber*

There’s nothing wrong with showing off what the VR system can do using these action-packed commercials. In fact, these companies would not exist if successful marketing campaigns such as these had not convinced thousands of people to buy VR systems in the first place. However, this advertising technique is inherently excluding people with motion disabilities. VR systems also offer games that do not require much movement, such as games where you explore a national park or museum. The Oculus Quest offers a “seated” mode to most games where users can relax in a chair while playing. These games do still require some hand movement as the user is expected to use the controllers to navigate and play, but these “seated” experiences rarely appear in the media. As a result, VR is portrayed as an experience made for people who can jump and wave around their arms as they fight virtual zombies and play virtual sports.

3.2 The Virtual Body

Before virtual reality, people controlled characters in video games using their fingers on analog sticks or controllers. In this way, controlling the virtual bodies was not intuitive -- the body may jump with a press of one button and swing a sword at the press of another, which is entirely unlike jumping or swinging a sword in real life. In a sense, this evens the playing field for everyone who can use the required controller -- the virtual body will jump the same height and swing the sword at the same speed regardless of the strength of the player. The virtual body knows no pain, experiences no fatigue, and fears no permanent death. It is, in a sense, perfect. All players who can use an appropriate controller can experience this subtle yet empowering perfection.

Virtual reality has no such even playing field as it demands the player to be the virtual body. Anybody who has played both 2D and VR first-person shooters knows the startling difference between the two. In 2D, a player merely swivels their thumbs and twitches their fingers to aim and shoot a hoard of attacking zombies. In VR, a player must raise their arms to aim the virtual gun. They must turn their physical body around to survey the area. They must experience the thrill of a zombie running toward them as it roars and swipes its claws through the air. They must push and punch the zombie away from their physical body to protect their virtual avatar. It can actually be quite scary for people who have never played this kind of VR game before! Bailenson wrote in his book *Experience on Demand*, "having watched thousands of people in my lab, many of them repeatedly experiencing VR, I can tell you that VR experiencers behave differently. It takes a lot of viewings and a lot of willpower not to flinch when the crates start falling in the earthquake simulator."[3] The environment can be so immersive that the player nearly forgets that they are only a headset away from escaping danger. They may scream in playful terror, punch and shoot and jump in a desperate effort to fight off their

[3] Jeremy Bailenson, *Experience on Demand,* p. 601.

attackers, all the while flailing around their physical body. It is quite the spectacle. In this way, VR is a gaming experience unlike anything a 2D game could deliver.

Virtual reality with its life-like mapping to the virtual body is, therefore, a much more intuitive design. What could be a more obvious way to throw a virtual punch than to throw a physical punch? This is some of the delight of VR—the satisfaction of completing virtual actions using full-body movements in the physical world. People who were never terribly impressed by 2D gaming could be drawn into the immersive, intuitive motions of VR design.

This new approach to controlling the virtual body comes at the greatest cost to accessibility yet, and the most uneven playing field known to modern gaming. The virtual reality body is the same regardless of who wears the headset in that it occupies space in the same way. But real bodies do not all occupy space in the same way. Real bodies have an infinite number of differences, all of which need to be mapped into the same virtual body. This is, of course, a problem for those with motion disabilities, as there are assumptions about how the virtual body moves are how the player completes those movements.

Virtual reality technology is still in its infancy and will likely become more adaptable, versatile, and accessible over the coming decade. However, there is no need to wait for the technology that could make virtual reality experiences hands-free – it already exists.

3.3 Virtual Reality is Already Hands-Free

Virtual reality headsets are already equipped with the technology to be hands-free devices; they only lack the interface design necessary to be used without hand controllers (an example of a hands-free VR interface will be discussed shortly). Gyroscopes in the headset help it detect motion and microphones listen to the user's voice. These sensors are already enough for a hands-free gaming experience, but some headsets include even more advanced technology. The HTC Vive Pro Eye has the Tobii eye-tracking system built in for optimized VR experiences. The 2021 HP Omnicept Edition VR headset will have face-tracking, eye-tracking, and a heart rate monitor. These extra sensors are intended to determine when the user is overwhelmed, but they can be used for other purposes as well.

This is why VR is quite promising technology for the future of hands-free gaming – no further inventions are required. This is also why my thesis will focus primarily on head gestures rather than gestures or movements from other parts of the body. This paper was written with the future of gaming in mind, considering the current state of gaming technology.

3.3.1 Current Hands-Free VR Experiences

Virtual reality headsets are already helping people access hands-free experiences, even if indirectly. *Arca's Path VR* is a hands-free virtual reality game in which the user guides a ball through a puzzle maze using head motions. The game offers 25 levels of challenging physics-based gameplay, beautiful crystalline landscapes, and an ambient soundtrack of chilled experimental electro music.

Arca's Path VR provides a delightfully immersive and engaging hands-free experience that does not feel lacking or limiting without hand controls. While not as complex as some virtual reality adventure experiences, this game still provides a captivating and aesthetically-pleasing escape with more interaction than other popular hands-free experiences, such as watching a movie. *Arca's Path VR* is a good reminder that the goal of a hands-free gaming experience should not necessarily be to produce a highly complex yet still accessible game, but a game that balances optimal user experience with challenge.

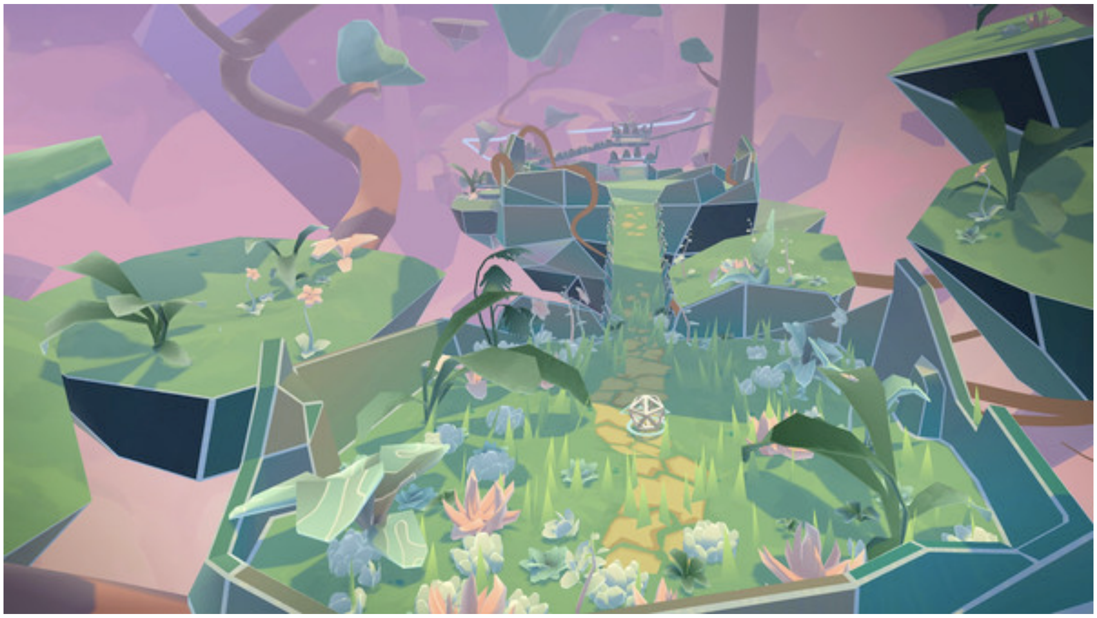


Figure 3: *Arca's Path VR* Screenshot from its Steam profile page

The *Eyeflite Ava* is a hands-free virtual reality software that can be accessed through the Oculus Go headset. The software provides the user with wireless private computer access that allows web browsing, a text editor, and text messaging. Navigation only requires head movements (or "head gaze controls"), and the user may choose to use voice dictation for typing. There are even a few head motion mini-games and relaxing environments to explore.

This software is inspiring for hands-free interface design because it demonstrates how head motion navigation is achievable with current virtual reality technology. The interface relies on using "dwell," or staying still, to make selections and head movements to move a cursor over the interface. *Eyeflite Ava* is especially interesting for the sake of this thesis because it is specifically designed for people with motion disabilities. The headset has features meant to best serve people with disabilities, such as notifying a caretaker when the user wants the headset removed.

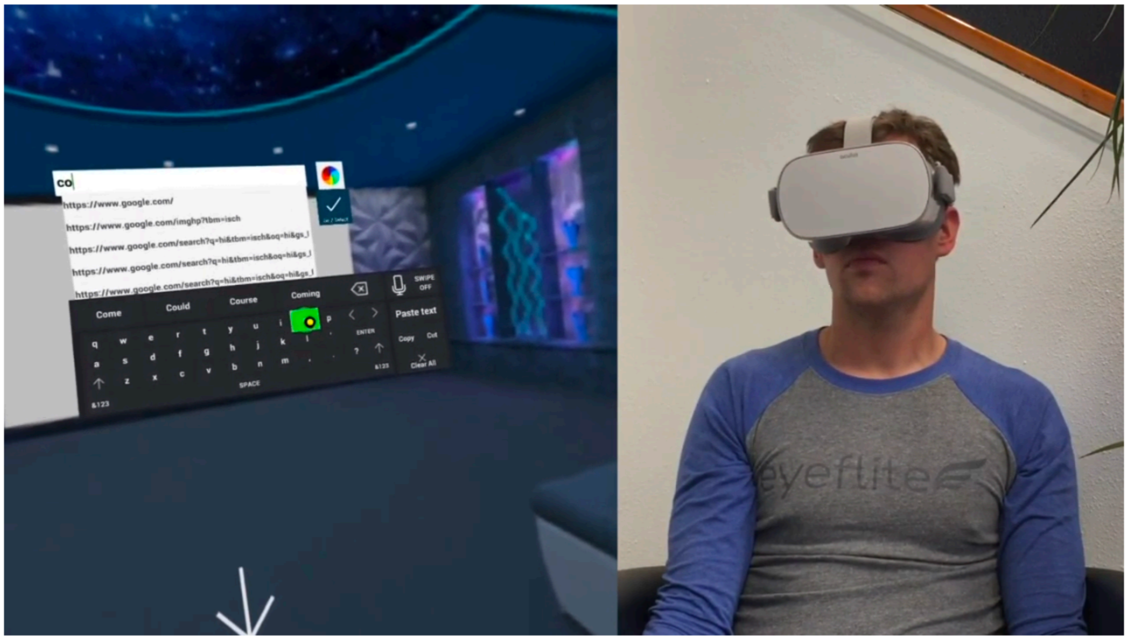

co
https://www.google.com/
https://www.google.com/imghp?tbm=isch
https://www.google.com/search?q=hi&tbm=isch&oq=hi&gs_l
https://www.google.com/search?q=hi&tbm=isch&oq=hi&gs_l
https://www.google.com/search?q=hi&tbm=isch&oq=hi&gs_l
Come
Could
Course
Coming
SWIPE OFF
Paste text
ENTER
Copy
Cut
Clear All
SPACE
&123
eyeflite

Figure 4: ***Eyeflite Ava*****, screenshot taken from the "Intro to Ava" YouTube video**

3.4 Better VR for Everyone

Unlike other gaming technology, virtual reality headsets sit on the user's face. Headsets weigh about one pound or more and press firmly on the forehead, nose and cheeks. It is not difficult to imagine how this may become uncomfortable over time—most people are not used to wearing heavy objects on their face. Further, the discomfort from the device itself is by far the least of VR comfort issues.

The biggest problem facing VR comfort is motion sickness. This is especially true in games where the player "travels" farther in the virtual world than they do in the physical world. Some games remedy this problem by not showing the player in transit and skipping immediately to the destination after a "blink" of darkness, but this does not work for everyone. Players also complain of eye strain, headaches, fatigue, acne (from poor headset hygiene), disorientation, injuries and even seizures.

There is no one fix to making virtual reality more comfortable for users. This is partially because there are so many factors leading to the above problems. The single biggest problem is latency, which "in most VR systems causes more error than all other errors combined."[4] Latency means there is a delay between the user's action and the machine's reaction. This is a huge contributing factor to motion sickness. Others are a lack of precise calibration, low accuracy in head-trackers, low refresh rate, and low tracking precision leading to jitter or shaking. Other problems include an offset center of mass which can tire the neck, an improperly fitting headset that could lead to headaches and excessive pressure on the face, and misaligned distances between lenses and other imaging parts that lead to visual discomfort symptoms.[5]

One problem with VR headsets is that they prevent your eyes from focusing on objects naturally, leading to eyestrain. In everyday life, eyes perform two tasks to focus – converge and accommodation. Converging means that the eyes move together to focus on objects. The eyes point straight when looking at far away objects and slowly tilt inward to be more "cross-eyed" as they focus on closer objects. Meanwhile, accommodation happens individually, meaning each eye changes its shape to focus

[4] Jerald, Jason. *The VR Book: Human-Centered Design for Virtual Reality*, p.. 197.
[5] Jerald, Jason. *The VR Book: Human-Centered Design for Virtual Reality*, p. 198-200.

appropriately. When performed correctly, this means that only what the eyes are looking at will appear sharp and clear to the person. Objects in front or behind will blur. In VR, converging occurs normally as the eyes look into the headset lenses, but accommodation is stunted. The focus of the lens is predetermined by the creator of the headset, and therefore no amount of accommodation will affect the clarity of the scene. Eyes may work exceptionally hard to accommodate to no avail, leading to fatigue.[6]

The problem here is that the VR headset has no sensor to determine what the user is focusing on, so it cannot adjust the scene's sharpness accordingly. While depth perception sensors exist, you are unlikely to find one outside of an ophthalmologist's office. But imagine if VR headsets included these sensors – how so much more space is opened up for user interaction. Imagine accessing menus simply by focusing on them, or jumping in and out of applications by focusing on their icons. Even in crowded environments, the user could navigate between actions simply by looking at where they are in space. This would be a massive leap for hands-free design, as it could be paired with eye-tracking software for faster and more precise navigation, typing, gameplay, etc. Changing focus could be a hands-free way for a user to select a button or back out of a menu.

[6] Jeremy Bailenson, *Experience on Demand,* p. 948 – 960.

Adding a depth-perception sensor would not only make VR more accessible, but a more comfortable experience for all players. The same could be said with improving some of the existing sensors. More adjustable and lower-weight headsets would make VR more accessible to people with head and neck pains as well as people whose faces or skulls do not conform to average dimensions. Higher precision and accuracy in head-trackers will improve hands-free navigation as well as reduce motion sickness. Face-tracking could be used to adjust settings based on sensing user discomfort as well as be used to control the interface. Focusing on creating a better hands-free experience will consequently optimize the experience for all players.

4. Analysis of Controller Structures and Alternatives

This thesis is an exploration of a hands-free game design. Until this point, I have been explaining all the background information that led me to this concept and the driving forces behind my decisions. In short, games are a massive part of modern culture, capable of bringing feelings of joy, accomplishment, and empowerment, which is why I believe that games are more than activities for children and should be taken seriously in terms of inclusivity. Modern video games are seemingly designed for "anyone" to play, but this is not the case as they require the use of two hands. This is why I focused on a hands-free design – it is not just for people with motion disabilities, but for people in all kinds of situations where they cannot use their hands. Virtual reality technology is already equipped with all the sensors to make gaming hands free, which is why I have decided to write about a head-gesture-centric design – there are no further inventions to be made, just software to be written.

This next chapter of the thesis and moving onward I will be describing my hands-free gaming concept. The concept is a multi-step streamlined experience for hands-free gaming that starts with the game designer and moves to the gamer. I will propose a method of categorizing game controls by frequency of use that could make controllers more accessible and easier to remap. But first, I will cover background information on possible hands-free gaming gestures and alternatives.

4.1 Head Motions and Gestures

There are numerous motions and gestures used to operate computers and replace hand controls in video games. A few major categories are covered below.

4.1.1 Degrees of Freedom

The six degrees of freedom (6DOF) refer to the possible movement of an object in three-dimensional space. Imagine an object floating freely in space — it may change position by moving up/down (heave), left/right (sway), or forward/backward (surge). It may also rotate in space on the normal axis (yaw), transverse axis (pitch), and longitudinal axis (roll).

The three degrees of freedom (3DOF) commonly referred to in the context of virtual reality are the rotational motions pitch, yaw, and roll. The 3DOF are the ways in which a person's head can move without moving other parts of the body, making the 3DOF especially relevant to hands-free virtual reality gaming. The easiest way to imagine these motions is roll as a nod, yaw as shaking the head "no," and pitch as a confused head tilt. These are the major head gestures that assist in navigating virtual reality interfaces without using the hand controllers.

The following diagram was taken from Wikipedia.

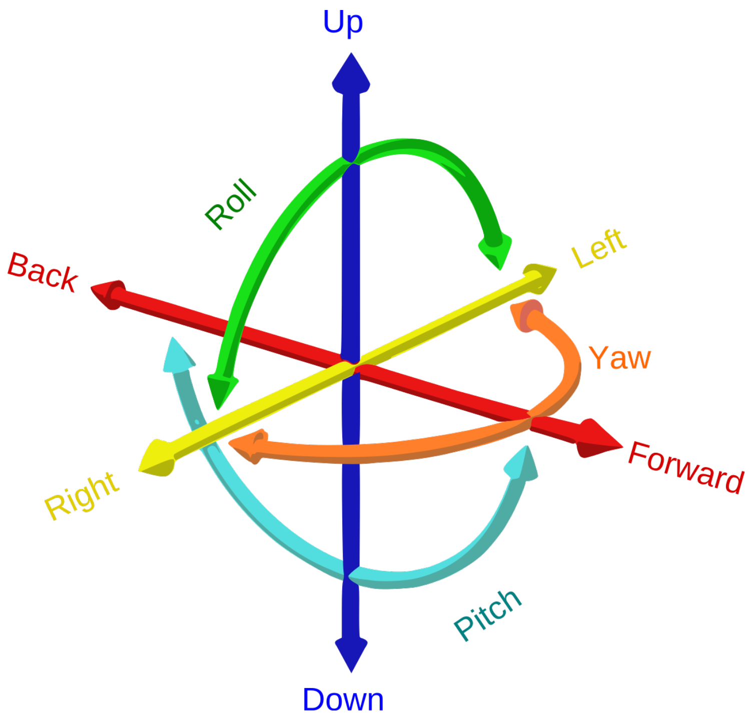


Figure 5: The Six Degrees of Freedom

4.1.2 Eye-Tracking

Eye-tracking technology uses cameras pointing at a person's eyes to determine where they are looking. The most accurate trackers use four cameras, with two pointed at each eye, but some eye-tracking can be accomplished with a single camera and a few minutes of calibration. State of the art eye-tracking technology uses four infrared cameras and shines near infrared light into the eyes for high-precision tracking. Infrared cameras are better because a person's pupil glows in infrared light, making it very easy for the computer to track its movements.[1] This is demonstrated in the image below, taken from Wikipedia.

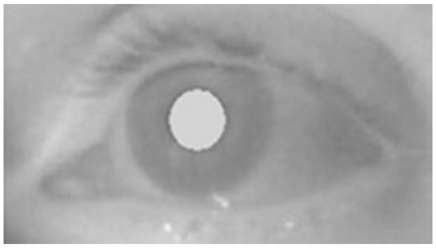


Figure 6: Pupil glowing in near-infrared / infrared light

Eye-tracking has long been used as a possible hands-free alternative for controlling computers. The mouse is moved with the eyes and clicking is accomplished through other gestures, such as speech recognition, dwelling on a selection, or blinking. Some companies, such as Tobii Dynavox,[2] have created software that considers a 3D model of the eye. The 3D perspective increases the quality of the user experience by allowing the user to move their head freely without disrupting the accuracy of the eye-tracking.

Eye-tracking has proven to be a powerful tool for those with motion disabilities, but has limitations for its use in the gaming universe. The most glaring problem is that people do not necessarily want to look at something to select it. Eyes move more than three times per second on average. People constantly use their eyes to explore their

[1] There are actually two general types of near-infrared eye-tracking techniques depending on the location of the illumination source: bright-pupil and dark pupil. Bright-pupil is used more often because it creates a larger contrast between the iris and pupil.

[2] Tobii Dynavox website link: https://us.tobiidynavox.com/

surroundings, and this is no different in gameplay. *Pac-Man* players may wish to look around the screen without Pac-Man following their eye movements. *Pong* players may want to look across the screen without their eye movements controlling the placement of the paddle. For this reason alone, eye-tracking can only be useful in certain gaming contexts, and likely will need to be accompanied by an "on" and "off" switch.

Other limitations of eye-tracking include problems with precision, especially in 3D spaces. In a 3D space, only technologies created to track depth perception can determine what exactly a person is looking at. Eye-tracking technology is also less accessible because high-quality glasses are expensive. There is a wide range of prices, but most are over $1,000 and some over $10,000. Free software that uses the computer's built-in webcam is extremely limited in accuracy and use cases. Users who are serious about using eye-tracking as a means of controlling a computer mouse are best off finding at least two cameras and subscription software for eye-mouse control.

4.1.3 Face-Tracking

Face-tracking is another possibility for controlling a computer mouse or other controller. Face-tracking software works by using databases of face images and geometric algorithms to determine the movements of facial features. These software packages are surprisingly comprehensive, tracking eyes, eyebrows, cheeks, and lips, and understand facial expressions such as frowning or sneering. Face-tracking has become widely popular for entertainment purposes -- Snapchat and Instagram use this capability to apply amusing filters to people’s pictures and videos, such as cat ears, glasses, and facial hair. Apple uses face-tracking for everything from creating emojis that match one’s expression to using Face ID to unlock one’s iPhone.

Face-tracking controllers work in a variety of ways. Some, such as Kinesic Mouse,[3] are completely customizable -- a user can select which facial movements or expressions should be mapped to which controls. Others are more simple, such as Smyle Mouse.[4] The user moves the mouse through head movements (the mouse seems to follow where the user’s nose is pointing) and the user smiles to click. Customizable options are more realistic for complex gameplay, but simple options may be better for basic computer use and perhaps casual PC games.[5]

Limitations to face-tracking are similar to that of eye-tracking in that face-tracking software can be imperfect, expensive, and unrealistic depending on the situation. While not as expensive as eye-tracking glasses, face-tracking software is usually several hundred dollars (as of 2020) and is often a subscription service, so this cost is reoccurring. Face-tracking relies on the user’s ability to have great control over their facial movements and focus on their expressions. If facial movements are tracked from gameplay control, then involuntary smiles, gasps, sneezes, etc. could disrupt play. Games where a player must press multiple buttons at once or in rapid succession would

[3] Kinesic Mouse website link: http://kinesicmouse.xcessity.at/index.php

[4] Smyle Mouse website link: https://smylemouse.com/

[5] See Appendix A for a closer look at the great accessibility of casual games

also be difficult to play using face-tracking (and perhaps impossible to play with eye-tracking). These are just a few of the problems with face-tracking technology as a gaming controller -- many more lie in the nuances of facial muscle fatigue, precision, etc.

4.1.4 Speech Recognition

Speech recognition has been a huge asset for hundreds of uses over the decades, and voice recognition technology has advanced dramatically since its inception. Speech recognition software and apps are plentiful, and many are specialized to help users with specific tasks, such as writing papers or translating to different languages. Artificially intelligent personal assistants are built into smartphones so users can search Google, send texts, and make calls while driving or doing other tasks that prevent typing.

There are several dozen programs that help users speak to control games in a variety of unique ways. MouseGrid divides the user's screen into nine numbered sections and the user says the section they want to select. Then that section is cut into nine sections and the process repeats. This is demonstrated in the images below, taken from Nuance.com. Some programs, such as Voice Finger, respond to literal key commands so the user can use the keyboard with their voice. Windows Speech Recognition Macros allows the user to create custom commands, including sequences of actions. These are just a few of the many available options.

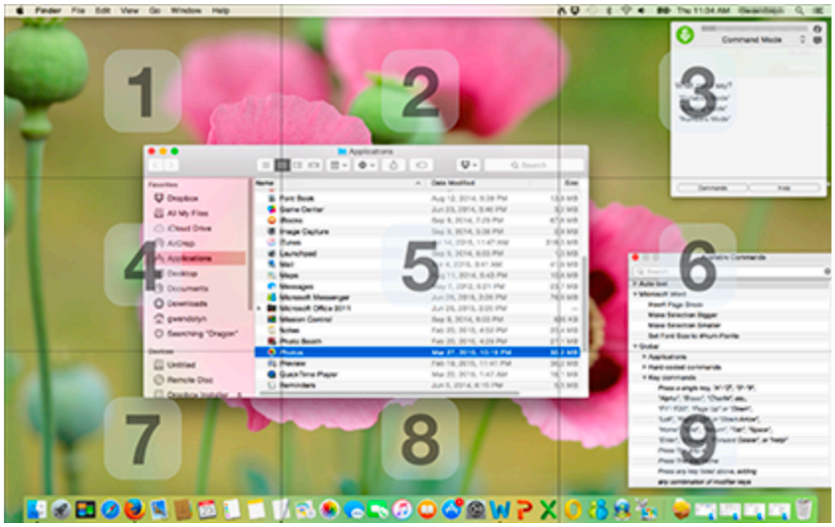
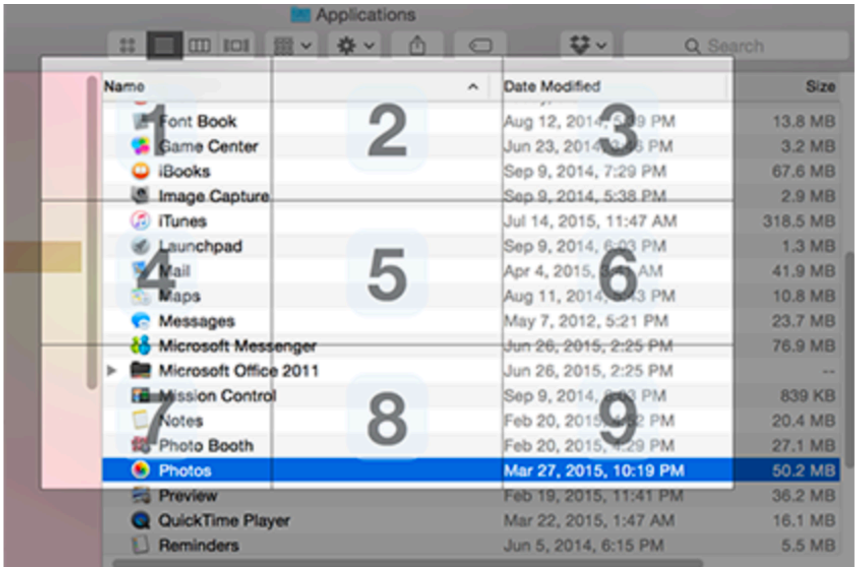


Figure 7: MouseGrid demonstration, part one



Name	Date Modified	Size
Font Book	Aug 12, 2014, [illegible] PM	13.8 MB
Game Center	Jun 23, 2014, [illegible] PM	3.2 MB
iBooks	Sep 9, 2014, 7:29 PM	67.6 MB
Image Capture	Sep 9, 2014, 5:38 PM	2.9 MB
iTunes	Jul 14, 2015, 11:47 AM	318.5 MB
Launchpad	Sep 9, 2014, [illegible] PM	1.3 MB
Mail	Apr 4, 2015, [illegible] AM	41.9 MB
Maps	Aug 11, 2014, [illegible] PM	10.8 MB
Messages	May 7, 2012, 5:21 PM	23.7 MB
Microsoft Messenger	Jun 26, 2015, 2:25 PM	76.9 MB
Microsoft Office 2011	Jun 26, 2015, 2:25 PM	--
Mission Control	Sep 9, 2014, [illegible] PM	839 KB
Notes	Feb 20, 2015, [illegible] PM	20.4 MB
Photo Booth	Feb 20, 2015, [illegible] PM	27.1 MB
Photos	Mar 27, 2015, 10:19 PM	50.2 MB
Preview	Feb 19, 2015, 11:41 PM	36.2 MB
QuickTime Player	Mar 22, 2015, 1:47 AM	16.1 MB
Reminders	Jun 5, 2014, 6:15 PM	5.5 MB

Figure 8: MouseGrid demonstration, part two

Speech recognition technology can also be trained to respond exceptionally well to a specific user. For example, Apple's built-in iPhone assistant Siri can be trained by the user to only respond to their voice. To "wake" Siri, a user may say, "Hey Siri." This feature was abused (although often innocently) by people who wanted to control their friends' phones. They might have said, "Hey Siri, text mom…" to send a text on someone else's phone without their consent and without having to unlock their phone. Burger King created a commercial that abused this feature on Google's devices, which respond to the phrase, "Okay Google." Burger King ran ads that asked, "Okay Google. What is the Whopper sandwich?" and let the users' devices do the rest, responding with, "The Whopper is a hamburger, consisting of a flame-grilled 4-ounce beef patty, sesame seed bun, mayonnaise, lettuce, tomato, pickles, ketchup and sliced onion."

Speech recognition's most glaring limitation is its terrible inefficiency. Understanding speech is slow -- a computer must parse words, interpret them, and then execute a command each time. When the wrong command is executed, the action takes more time to correct. Speech impediments, accents, and proper nouns all pose problems. Distinguishing between similar-sounding words is also still difficult for programs (think about "clothes," "close," and "closed"). The grid of nine sections is an extremely slow way to click around a screen compared to common mouse control. All this considered, speech recognition software is at least quite affordable. It is available on all operating systems for free or low prices (under $15 in most cases, as of 2020).

4.1.5 Sip and Puff Devices, Lip and Tongue Movement

There are a variety of lip and mouth control devices available for some level of computer or game control. The QuadStick uses four sip/puff pressure sensors and a lip movement sensor, which can be assigned to any game controller. This technology is specifically made for people with quadriplegia, but the sip/puff system easily extends to people with a variety of disabilities. Similar technologies are used to help people control wheelchairs and motorized scooters. The Quadstick comes in three different versions, meant to make the device more accessible to people with different abilities or lower budgets. The following image is of the Quadstick FPS game controller, taken from the company website.[6]

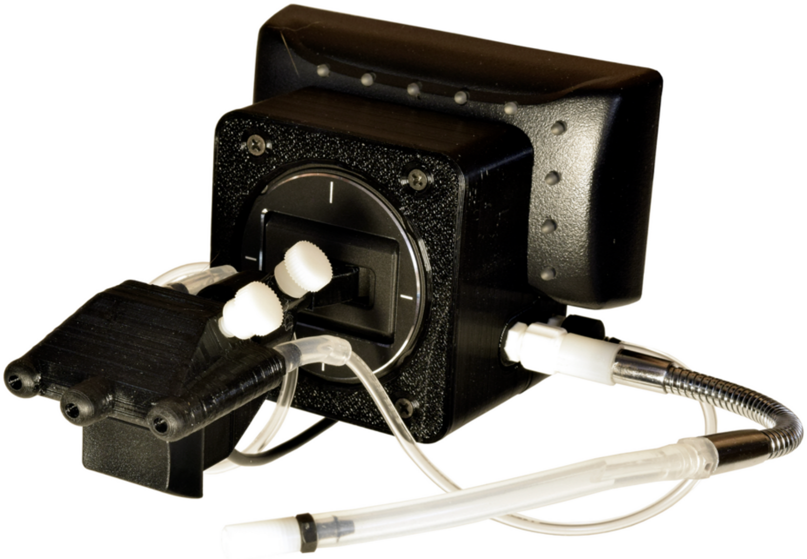


Figure 9: Quadstick FPS gaming controller

Some wheelchairs and other devices can be controlled by tongue movement, which has proved highly successful for those with spinal injuries or other conditions that cause wide-spread paralysis. In some iterations, the user wears a small tongue

[6] Quadstick company website link: https://www.quadstick.com

piercing that communicates with a headset. In others, the user wears a retainer that holds a sensor to the roof of their mouth. This sensor can detect tongue location and pressure to control a machine.

Very few companies have picked up sip/puff, lip or tongue movement systems for game control, making options limited for those who would like to try this method. Even the cheapest Quadstick is over $400. Customers must pay $55-$90 in shipping and may also need to purchase expensive converters, bringing the total to over $700. This technology may become cheaper in the future as new models are created and older versions become discounted. For now, players on a tight budget may need to opt for cheaper alternatives.

4.2 Breaking Down Game Actions by Frequency of Use

Ideally, the ergonomics of any gaming controller are good enough to allow the player to enjoy a few hours of gaming without any discomfort. Players who found a controller uncomfortable or painful wouldn't be likely to play for long, whether it is a hand controller or hands-free system. When considering a hands-free gaming experience, the first priority is comfort.

I will not attempt to determine who exactly would use a hands-free option from an abilities perspective. There is such a range of abilities and health conditions, much of which fluctuate in effects on a daily basis, that creating any one controller map would be minimally useful. Rather, the most useful controller would be the most flexible. It should be able to adapt to a variety of inputs, and ideally, players should be able to create and save their own maps for specific games. Multiple maps per game would be preferable because the same person may experience oscillating comforts and abilities on a day-to-day or even hourly basis.

It is necessary to implement different controller maps for different games because of the aforementioned priority of comfort -- different games will require heavier use of certain buttons or inputs. While this may not be a big deal for fully able players using common hand controllers, it could present problems for players with motion disabilities using specialized controllers.

For example, consider if the same hand controller is used to play the classic games *Pac-Man* and *Pong*. Let's say *Pac-Man* was mapped to the controller as so: the stick is used to navigate, and the A button is used to select options on the game menu. Let's say *Pong* was mapped to the controller like this: the directional pad is used to navigate, and the X button is used to select options from the game menu. Both of these games have two fundamental actions: navigate and select. But if the hand controller was strictly and inflexibly mapped to, let's say, a face-tracking controller, then the player would need to be able to comfortably complete four different actions (two kinds of navigation actions and two kinds of selection actions) as opposed to two. If the player could map the hand controller to the face-tracking controller however they would like for each game, the player could make both kinds of navigation the same facial gesture according to what is the most comfortable for them. This is especially important for people who have a limited amount of gestures they can comfortably complete in repetitions.

The problem is exaggerated with games that use many different buttons in various combinations, which is the most common scenario with modern games. Further, the same game may appear on numerous consoles and present slightly different controls on each one (these differences are the most apparent between the same game on a PC and a console).

To help mitigate the confusion around game controls and buttons, I propose that each game be broken down by the game designers into primary, secondary, tertiary, and quaternary actions based on the frequency of the action. The more frequently an action is used, the more important that it be a highly-accessible and comfortable gesture for the player. Not every game will have the same kinds of actions used at the same frequencies, which is why each designer must assess each game independently.

I encourage the designer to make these decisions upfront because it is the designers' job to shape video game accessibility, not the players. It is already increasingly popular for modern game companies to store their games' controls in an API (application programming interface), and sometimes this is accessible to the public so that controllers can be remapped. If those same APIs could deliver information on the frequency of use for each game action, people could make more informed decisions on how to remap their controllers. This is not only about helping gamers with disabilities, but a gesture toward universal design principles. Everyone could map their controllers for more comfortable or faster configurations if they understood which game actions were the most important going into the game.

I will now provide my proposed definitions for primary, secondary, tertiary, and quaternary actions. These terms could be used to start seeing game controls in a way that helps to map them to new technologies more easily. These definitions are based on the frequency of use, but frequency is also relative to the overall length of gameplay. These definitions are therefore flexible and stand as potential guidelines rather than strict rules.

4.2.1 Primary Actions

A primary action is a continuous action or one that is used so frequently that a player calls this action more than once per 5 seconds of gameplay. In *Pac-Man*, a primary action would be navigation because this is used constantly during play. An action that is not called constantly, but is called many times in short succession during certain (non-rare) aspects of gameplay could also be considered primary. An example of this will be explained in the case studies.

4.2.2 Secondary Actions

A secondary action is used frequently in gameplay, but not constantly. This action may be called once or twice per 20 seconds, or occasionally in small bursts. For adventure games, a secondary action is usually the “action” button used to pick up objects or interact with NPCs (non-playable characters, or computer-controlled characters players may encounter). A secondary action is most actions that are not used constantly, but are used multiple times per minute of average gameplay.

4.2.3 Tertiary Actions

A tertiary action is an action players use about once per minute or as infrequently as once every few minutes. Tertiary actions are not “rarely” used as they appear often enough to be essential to smooth gameplay. In an adventure game, a tertiary action may be jumping or perhaps using a weapon, depending on the nature of the adventure.

4.2.4 Quaternary Actions

A quaternary action is any action that may be considered “rarely used” or used as infrequently to once every 5 minutes to virtually never. Quaternary actions are never used in rapid bursts and may be some of the last controls that a player may learn or discover. They may be something relevant to gameplay such as checking the map, or they may be somewhat removed from gameplay, such as accessing an options menu or saving the game. An action that is used infrequently, but is used in rapid bursts when it is called, should be considered tertiary.

4.2.5 Case Study One: The Legend of Zelda: Breath of the Wild

The Legend of Zelda: Breath of the Wild is a 2017 action-adventure game created by Nintendo for the Nintendo Switch. The game gained massive popularity for its stunning graphics, compelling storyline and engaging game mechanics. It was chosen for a case study because of its popularity as well as its extensive adventure-game mechanics.

Primary actions in BOTW are similar to most open-world adventure games. The two most used actions are navigational: Move, with the left stick, and Move Camera, with the right stick. These are primary actions because they are used constantly throughout the game as the player controls the hero, Link, from a third-person perspective. Even when the player is not moving Link and is, for example, navigating the inventory, the two sticks are essential.

The only other primary action in BOTW is the Attack button, Y. Admittedly, this is a bit of an edge case between a primary and secondary action because players can effectively play the game for long periods without using Attack by avoiding and running away from enemies. However, Attack is essential to successful gameplay for all players who wish to complete the game's major quests. When players are attacking an enemy, the Attack button is used over and over in rapid succession. Enemies appear as frequently as multiple times per minute in BOTW. For these reasons, Attack is classified as a primary action.

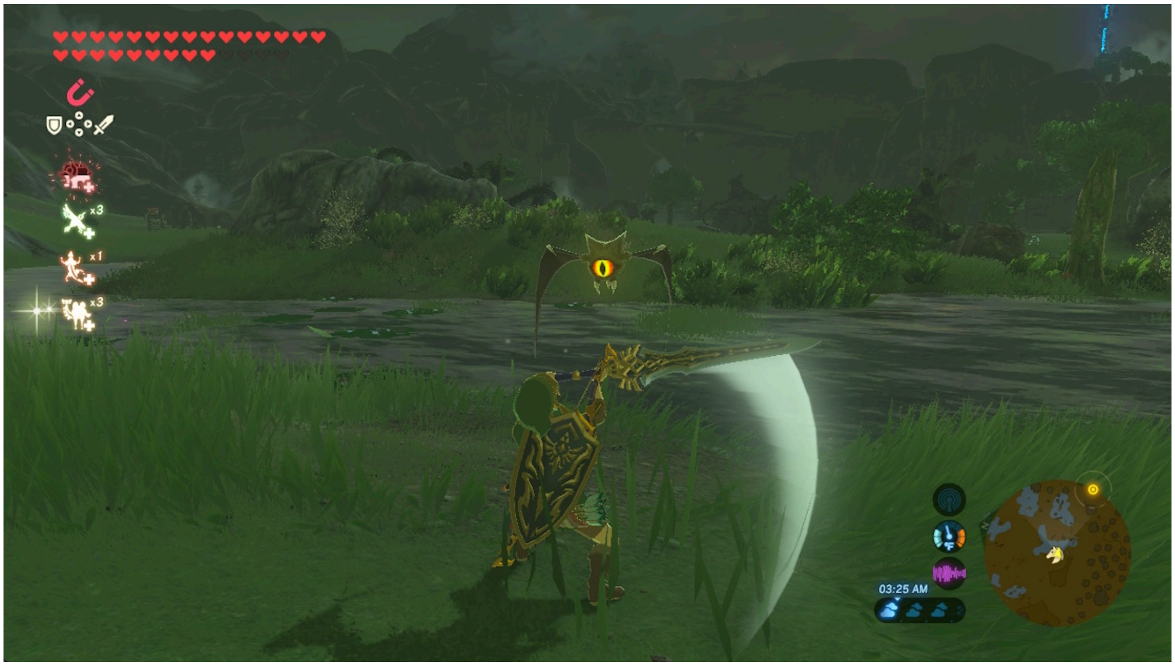
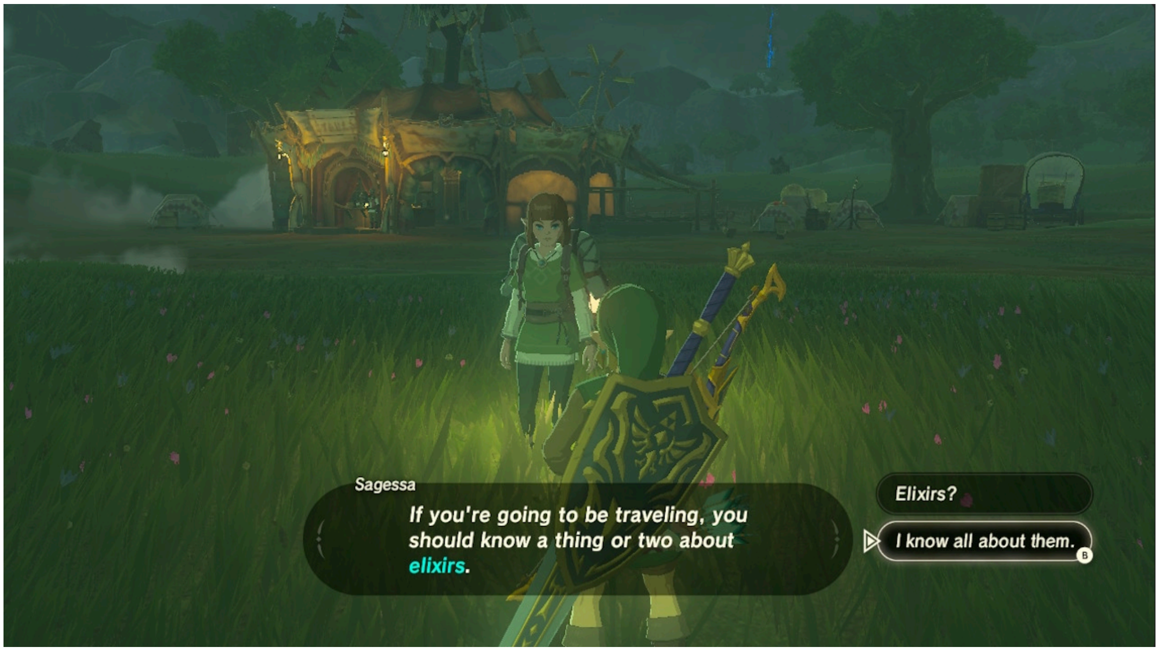


Figure 10: Link attacking an enemy with his sword

There are a few different secondary actions in BOTW. The most notable of which is Interact, or A. Players use Interact to pick up objects, talk to NPCs, and make selections in menu and inventory screens. Interact is seldom used in rapid succession, but is used frequently during gameplay. The next action is B, which doubles as both Cancel and Boost (but it will be called Cancel for simplicity). Cancel is the opposite of Interact in most ways. It is used to exit actions, end conversations, leave menus, and put items away. The same button is also used as Boost, which, when held, causes the character to run (as walking is the default) or swim fast (as wading slowly is the default). Boost can be held, but only briefly, and is not usually used more than a couple of times in succession, making it a secondary action. There are other secondary actions in the game but, for the sake of brevity, they won't be covered.

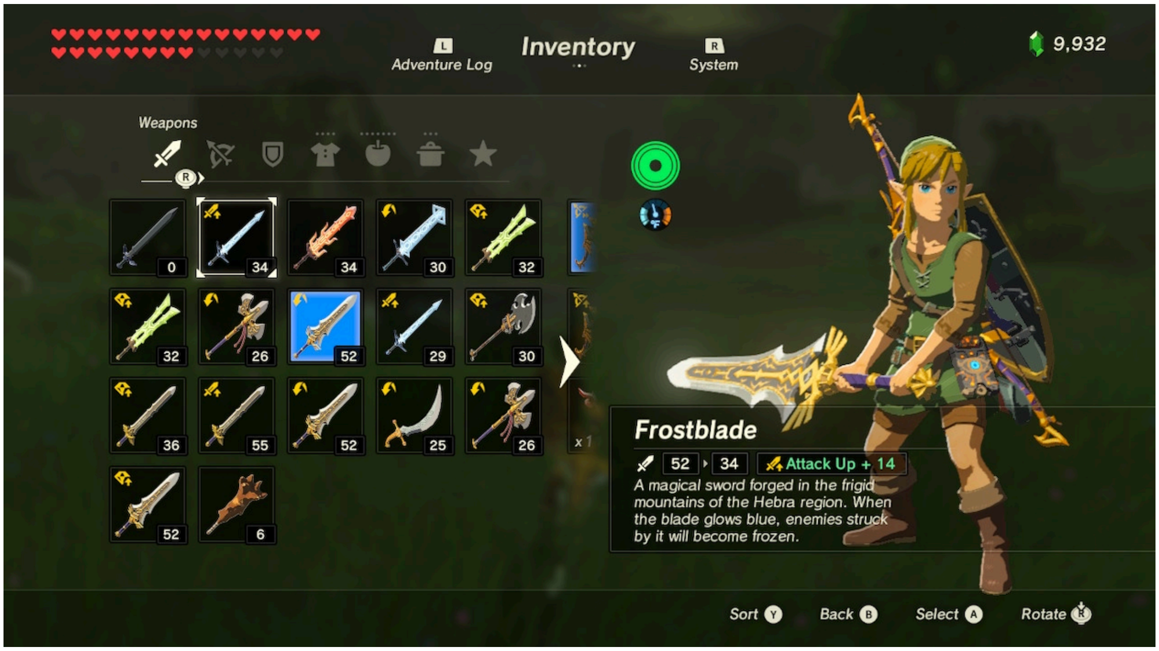


Sagessa

If you're going to be traveling, you should know a thing or two about elixirs.

Figure 11: How a player can use Cancel to end a conversation

There are more than a dozen tertiary actions in BOTW as many game mechanics are only used in specific, uncommon puzzles and situations. One of the more frequently used tertiary actions is the Pause button (+) which is a bit unique. Pause is used more frequently in BOTW than other adventure games because the Pause menu allows the player to access their inventory (food, clothes, weapons, shields, and more), view their quests, save the game, and tinker with settings. Players must access their inventory to replenish health, equip weapons, change clothes, and complete various other common gameplay chores. However, the Pause button is still only used a couple of times per minute at most and never in succession, making it a tertiary action.



A magical sword forged in the frigid mountains of the Hebra region. When the blade glows blue, enemies struck by it will become frozen.

Figure 12: The Pause / Inventory screen in BOTW

Quaternary actions in BOTW are actions that players can effectively play the entire game without. These include specialized fighting moves, such as Parry (ZL + A) and obscure gestures, such as Whistle (Down on the D-Pad). These actions are helpful but not necessary and are only used in specific, rare situations. Whistle, for example, is used to call one's horse from a short distance away.

4.2.6 Case Study Two: Mario Kart 8 Deluxe

Mario Kart 8 Deluxe is a 2017 car racing game published by Nintendo for the Nintendo Switch. The game gained massive popularity for children due to its exciting, colorful graphics, simple controls, and family-friendly themes. It was chosen for a case study because of the popularity of the racing game genre and the strong contrast in game mechanics from adventure-style games, such as BOTW.

The primary actions in *Mario Kart* are Accelerate (A) and Steer (Left stick). Because *Mario Kart* is designed to be played on a single-stick controller, either stick can be used for steering when using a single Joy-Con (or a mini, single-stick controller used with the Nintendo Switch). Both Accelerate and Steer are used continuously throughout gameplay, similar to driving a real go-kart in a race. Accelerate is held down throughout the race, and the same button is used to Select menu options, so it remains a primary button even when the player is not racing. Steering is constantly demanded due to the winding game courses, and the same stick is used to navigate menus, so it is also a primary action outside of racing.

While racing, players can obtain and throw special objects onto the racetrack to try to give themselves an advantage or slow down other players. Throwing these objects is a secondary action because they appear frequently on the course, and players are constantly trying to find items and determine the right moment to use them. Use Item (ZL) is also used to stop the Item Roulette, which spins before an item is awarded to the player.

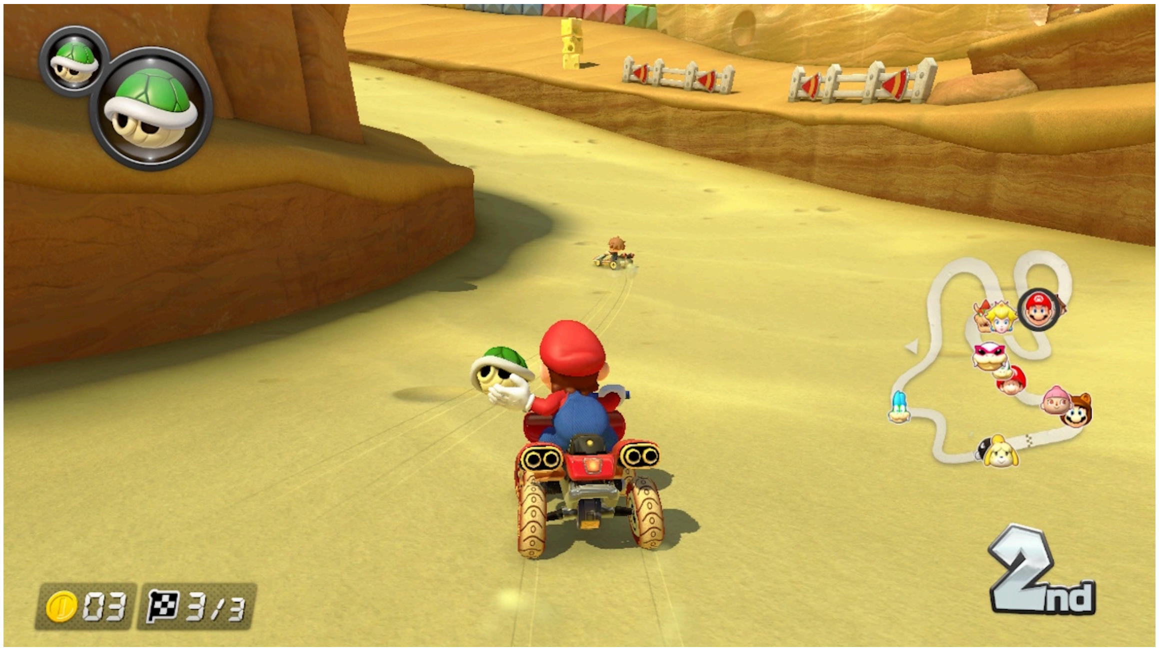


Figure 13: Preparing to Use Green Shell Item in *Mario Kart 8 Deluxe*

The tertiary action to be discussed is an edge case, depending on the player, and could be considered a secondary action. Drifting (ZR) can help players make sharp turns without slowing down. Skilled players drift as much as possible because they want to obtain the boost awarded after a sufficiently long drift. These players use Drift as a secondary action. Less experienced players (the majority of the fanbase) only use Drift on large or sharp turns, and some rarely drift at all. These players use it as a tertiary action. The same button is used to do a Trick after a player drives up a ramp and is airborne for a moment. Some race tracks offer this opportunity more than others, and some players take advantage of Tricks while others don’t.

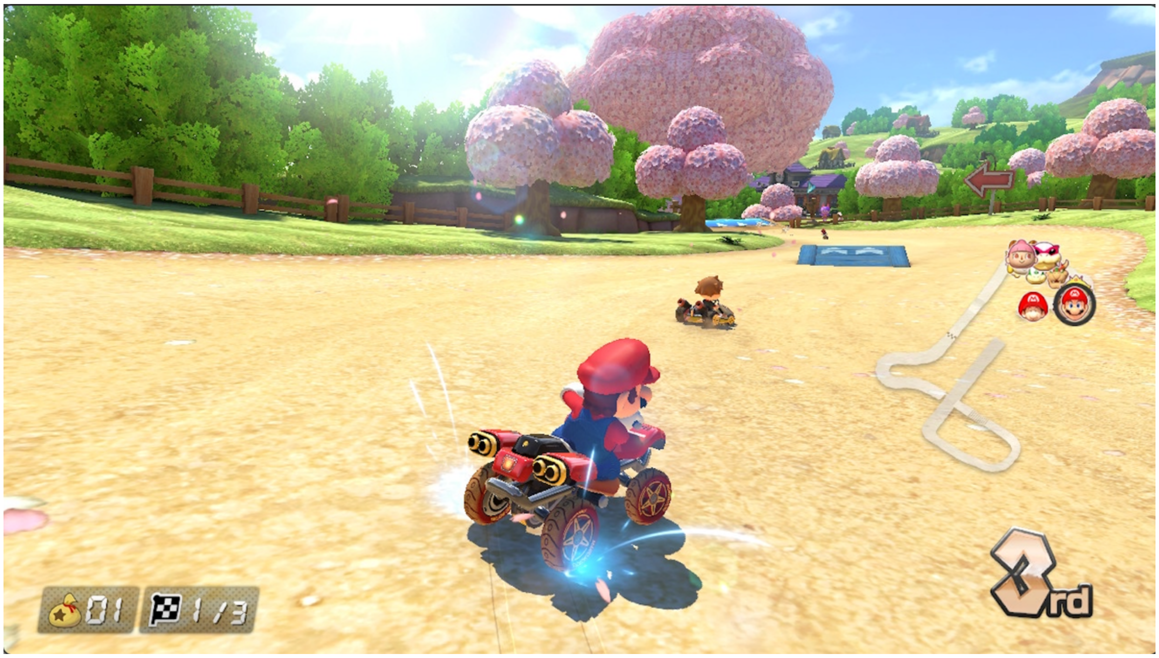


Figure 14: Drifting in *Mario Kart 8 Deluxe*

Quaternary actions in MK8D are actions such as Pause (+) and Look Backward (X). Neither of these actions is essential to gameplay. Look Backward is simply an animation and does not affect the racing. Unlike BOTW, there is no compelling reason for the player to access the Pause menu during play, other than to take a break from play. These unnecessary and rare actions are therefore quaternary.

4.2.7 Categorizing Controls Is Nothing New

Game designers have long understood the relationship between how frequently a button is used and user comfort. This is one of the many factors considered when a controller is being designed as companies strive to make the controllers as accessible as possible. In this sense, accessibility means that the controller is easy to learn and considers the limitations of people's hands and fingers.

I will briefly discuss the most compelling and relevant use of categorization I have found in my research. This categorization was found on game designer Andrew Dotsenko's blog.[7] Dotsenko describes the gamer's hand in terms of three sections. Primary control is for the thumb and index finger for their precision and flexibility, and such should be used for primary actions. He defines a primary action as "require active decision-making, the main 'Verbs'/basic mechanics that the player use. Might include more than one basic inputs (aim & shoot, moving & jumping, etc.). Require constant attention from the player." Secondary control is for the middle finger because it is "flexible but not so precise" and should be used for primary hold actions (or primary actions that need to be held down for a period of time). The final hand category is Support, which is the ring and pinkie fingers, as these are weak and inflexible. He says support fingers can be used for secondary actions, although he does not define these.

Dotsenko calls these categories "finger groups" and matches them with corresponding "tiers" on an Xbox controller. The controller buttons are broken into three tiers according to Fitt's Law: "the fewer the distance to the button and the bigger the button, the more accessible the button is."[8] Buttons that are the easiest for the player to press are Tier 1, while the hardest to press are Tier 3. The following figures are from his blog describing this methodology.

[7] Dotsenko, Andrew. "Designing Game Controls." *Game Design Framework.*

[8] Fitt's law is a model used to predict human movement, commonly used in the design of human-computer interaction and ergonomics.

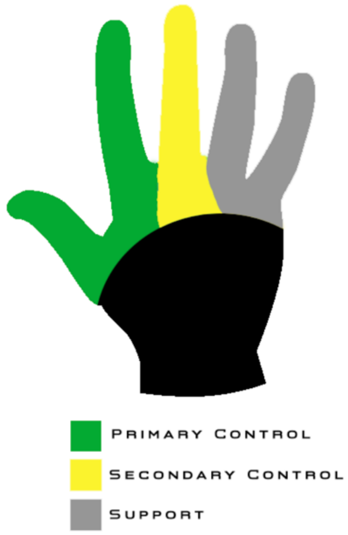


Figure 15: Dotsenko's Finger Groups

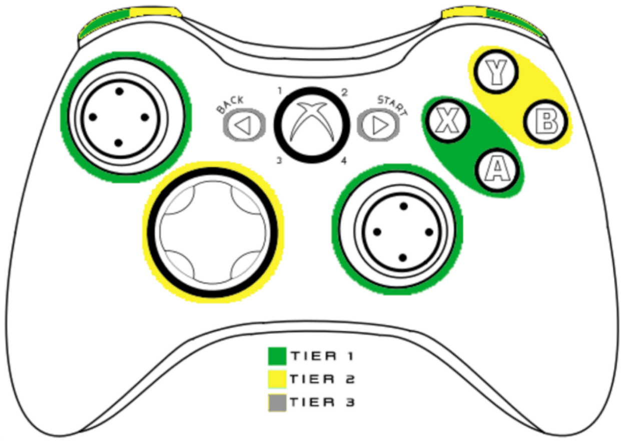


Figure 16: Dotsenko's Xbox button tiers

Dotsenko goes on to describe various ways to group game actions and controls, but for the purposes of this paper, these will be left out. The main point is that Dotsenko’s categorization is another way to think of breaking down game controls by their prominence or importance to gameplay. His example is relevant to hand controllers while mine is relevant to head gestures or hands-free controllers, which is why his ideas (and those of many others) are not extensively covered in this paper. While controller button categorization is nothing new, most of these concepts focus on controller ergonomics instead of alternative gestures.

4.3 How Controllers Changed Over Time

Earlier in this document I described how video game controllers have become more complex over time, and therefore, more exclusionary. Gaming controllers have evolved from a box with a single stick to a two-hand controller with over a dozen buttons designed to be pressed in up to 4-way combinations. However, this is not the only way that controllers have changed -- controller designs have also become more neutral with time as they have strayed from the assumptions that players will be pressing certain buttons more than others. Nintendo's controllers present the best example of this transition. I will focus on one jump in particular -- from the classic GameCube controller to the Nintendo Switch Pro controller.

The GameCube controller was released in 2001 to pair with what was, at the time, the newest Nintendo console. It has all of the same buttons as the Pro controller, minus three (the plus and minus buttons and the screen capture button). The missing buttons are not the most striking difference between these controllers -- it's the design. Below are side by side images of the two controllers. These photos were taken from Wikipedia.

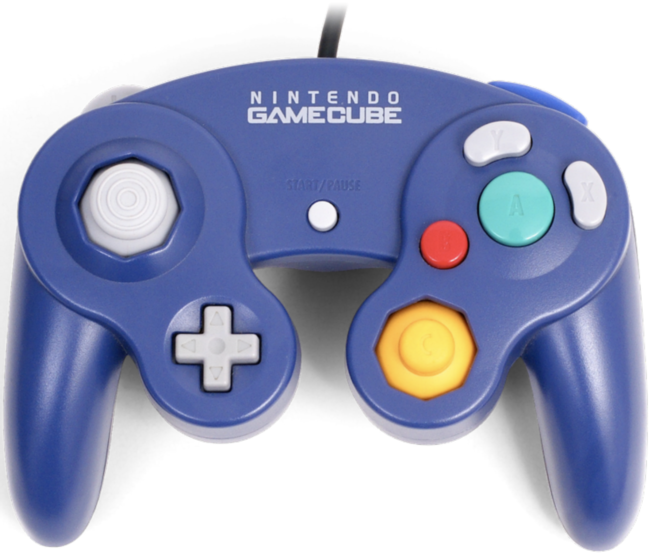


Figure 17: GameCube Controller, 2001

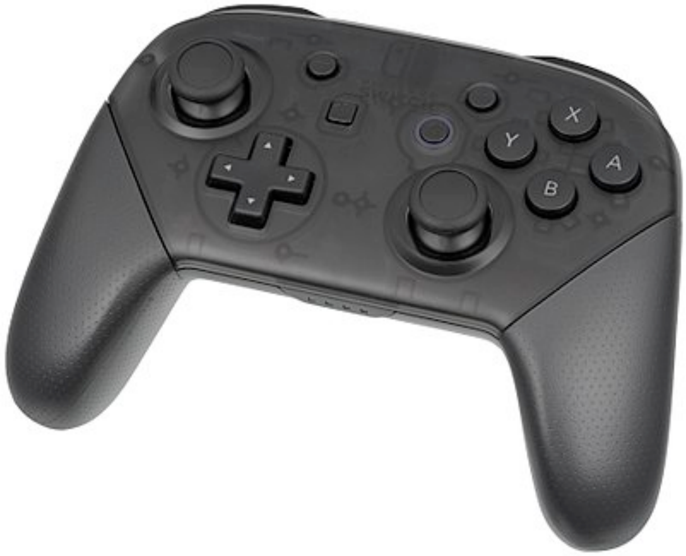


Figure 18: Nintendo Switch Pro Controller, 2016

There are several clear differences the player will notice right away -- the GameCube controller is colorful, with each button presenting as a different shape, size, and color. The Nintendo Switch Pro controller is muted in comparison -- all black with uniform buttons and sticks in shape and style.

The design of the GameCube controller suggests that certain buttons will be used more frequently than others. The same is true with the Pro controller, although to a lesser extent due to the lesser variation. Looking at the GameCube controller, which controls appear to be primary controls? One may guess that the large green button (A) and the large grey stick on the left would be primary controls, intended for heavy use. What about secondary? One may guess that the red button (B) could be a secondary control, and perhaps the grey bean-shaped buttons (X and Y) are either secondary or tertiary controls. GameCube games were made specifically for the GameCube and therefore were designed with this controller in mind.

Could similar assumptions about frequency of use be made for the Pro controller? Possibly, but to a lesser extent. One may assume that the left stick appears to be a primary control, but what about the cluster of buttons on the right? Perhaps A and B are closer to the player's thumb than X and Y, but is there any other way to distinguish them?[9] One could only guess that buttons such as A, B, X, and Y are more frequently

[9] The confusion of the A, B, X, Y buttons is amusingly demonstrated in this 2021 Reddit post, where a man taped a diagram of the controls to the TV screen so his wife wouldn’t have to look down as often: https://www.reddit.com/r/gaming/comments/l5r70v/my_wife_has_been_learning_to_play_video_games/

used than the Plus and Minus buttons. However, the easily-drawn conclusions stop around there. This neutrality in design suggests that Nintendo no longer wished to make suggestions about which buttons should be used for primary actions, secondary actions and so on. But the lack of neutrality in the GameCube controller design suggests that, at one point, they did.

So why does any of this matter? When Nintendo gave more freedom to game developers regarding which buttons would be most important to access for their games, they loosened some of the uniformity of video game controls' design. Almost any button could comfortably accommodate a primary action. Games could have numerous primary actions composed of more complex combinations. But this increase in button neutrality comes with a few downsides. Players are expected to memorize the placement of the A, B, X, and Y buttons because they are no longer distinguishable without reading the print on the buttons. Games may demand players to command a greater number of buttons and combinations in less predictable groups, making games harder to learn (The GameCube has 9 buttons, while the Pro controller has 12). Finally, remapping game controls to other devices is a task that must be completed one game at a time, as all games do not use the same or similar button combinations.

Due to the diversity and complexity of game controls in modern games, the most practical approach to creating a hands-free control system is to allow ample player customization for every game. This path also takes future technologies and controllers into account, as controllers may include even more buttons with time.

4.4 Non-Binary Controls

Most of the buttons on a standard game controller are binary, meaning they are either "pressed" or not, and there is no meaningful input in between. Some gestures could be seen as binary as well. For example, a person's eyes are either open or closed. A person is either looking at an object or they are not. A person's mouth is either open or it is not. For the sake of gaming controls, pairing binary controls to binary gestures works well. However, there are a few controls that are non-binary and therefore will require gestures that are also non-binary.

4.4.1 Full Two-Dimensional Controls

Sticks are the most prominent non-binary controls on standard game controllers. Sticks move in a full two-dimensional circle and are sensitive to both the direction they are pushed and how far in the direction they have been pushed. This is how games allow characters to move slower or faster depending on how far the stick is pushed and to move anywhere in a 360-degree circle.

Some gestures are better than others at emulating the full range of movement for sticks. For example, it would be quite challenging to capture a stick's full range of motion with one's cheeks or eyebrows. It may be easier with one's lips, and for the right game, eye-tracking is a possibility. Perhaps the most intuitive option would be controlling the stick using head motion through tilting and rolling.

Regardless of the player's choice, the nature of the two-dimensional controls should be considered when remapping controls to head gestures. This may seem obvious, but primary controls cannot simply be randomly assigned to a player's most comfortable or highly-preferred gestured. The nature of the control is just as important as the frequency categorization, if not more so.

4.4.2 Other Analog Controls

The other standard controls of potential concern are the trigger buttons, commonly known as ZL/ZR (Nintendo) and LT/RT (Xbox). These buttons usually operate like binary controls, but some controllers (especially Xbox controllers) have smooth triggers that do not "click" from a binary press, like most buttons on the controller. The triggers are analog and can be pressed slightly, half-way, fully, and anywhere in between.

Most games do not use the triggers like analog buttons. This may be because not all controllers are built with analog triggers, and game developers want their games to be playable on different consoles (which will require different controllers). However, games that do have analog triggers need appropriately matched gestures for maximum playability. For example, people can smile slightly or smile widely. They may squint or they may open their eyes wide. These are examples of gestures that could be used to emulate analog controls in the rare case that they are needed. Because this thesis does not include exhaustive research of all gaming controllers, it is possible that other controllers have similar analog controls that I have not mentioned. Each of these will need its own consideration to determine how it may be performed the most accurately with gestures.

4.5 Controller Map Breakdown: Nintendo Switch Pro

The Nintendo Switch Pro Controller was released in 2017 as a wireless controller to pair with the Nintendo Switch. It has 16 buttons and two sticks (which can also be pushed down like buttons, so perhaps that makes 18 buttons). This controller is the more ergonomic option over the smaller Joy-Cons. It is designed to be held with two hands with fingers positioned in what is sometimes referred to as a claw-like grip.

The controller is designed to be played with four fingers -- the thumb and index finger of both hands. There are six buttons and one stick available to each thumb and two buttons on the top of the controller available to each index finger. It is rare for a player to be expected to press more than one button with a single finger at the same time. For this reason, the controller can be broken into four quadrants: Upper Left (left index finger), Lower Left (left thumb), Upper Right (right index finger), and Lower Right (right thumb). This design suggests that players are expected to use a maximum of four controls at once.

When remapping controls to a new device or gestures, these four quadrants should be kept in mind to maximize playability. One type of gesture should be reserved for each quadrant. For example, just as a player would not be expected to press A and Y simultaneously, a player should not be expected to make two gestures with the same muscle at the same time (such as two lip movements or two head movements).

The quadrants need to be considered in conjunction with the primary, secondary, tertiary and quaternary controls of a specific game. These two factors will determine the most comfortable way for any one player to participate in a game. Each individual and game will be a unique pairing of abilities and controls to assemble a controller map.

To make my points clear, I will provide an example controller map with an imaginary example person. The example person, named Ava, will be using a controller mapped for *The Legend of Zelda: Breath of the Wild*, discussed earlier. Ava needs a hands-free controller and will be using face-tracking and speech recognition technology for gameplay. Ava is the most comfortable using lips, cheeks and head movements. She is fairly comfortable using her eyes. She has some comfort using her voice but prefers not to use it often.

The following is an example of a control map that Ava could try for *The Legend of Zelda: Breath of the Wild.* This controller map bears in mind that the Plus (+), Minus (-), Capture and Home buttons are virtually never used in conjunction with something else. In a sense, they are excluded from the concept of the four quadrants because they are used for more specific actions. The controller photos were taken from the official Nintendo website.

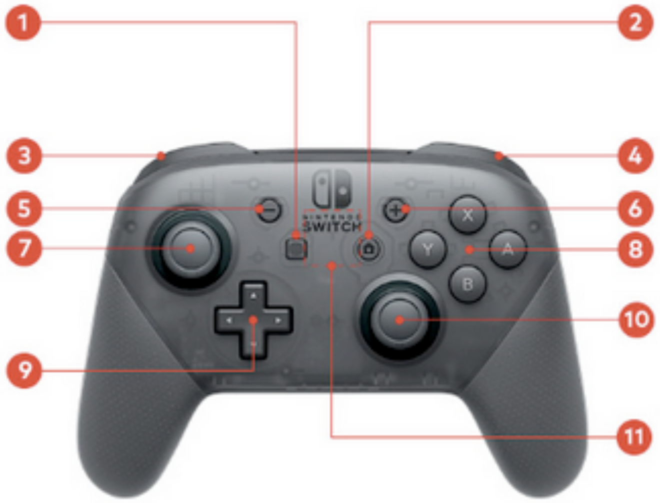


Figure 19: Nintendo Switch Pro Controller Map, Front

Note that for the sake of brevity, this table does not include every possible action from the game. There are many multi-button actions within the game, and none of those will be covered here. Also, the non-binary controls in this game are the two sticks, and both are paired with non-binary gestures. Below I have provided a brief explanation for what each of the columns means.

Action: The action performed in game upon pressing the corresponding button on a Nintendo Switch Pro controller.

Button: The button on the Nintendo Switch Pro controller that is used to call the correlating action. If the button is non-binary, it will be labeled with “NB.” Buttons without this label are binary.

Frequency: Based on the frequency of use, each command (button / action combination) is assigned to be a primary, secondary, tertiary or quaternary action.

Quadrant: States where the button is on the controller according to the four quadrants. A few buttons are marked “solo,” meaning they are virtually never used in conjunction with other buttons, and therefore do not need to conform to a specific quadrant.

Gesture Comfort: The user’s level of comfort performing the suggested gesture. Keep in mind, “low,” “moderate,” and “high” are relative terms. A “low” level of comfort need not mean that the gesture is genuinely uncomfortable, just not preferred.

Gesture: The suggested gesture to call the action instead of using the controller button. This is based on the user’s comfort with each of the available gestures, the quadrant the button is in, and the frequency in which that button is used.

Table 2: *Breath of the Wild* Hands-Free Controls Example (Front)

#	*Action*	*Button*	*Frequency*	*Quadrant*	*Gesture Comfort*	*Gesture*
1	Capture	Capture	Quaternary	Solo	Low	Say "Capture"
2	Home	Home	Quaternary	Solo	Low	Say "Home"
3	Use Rune	L	Tertiary	Upper Left	Moderate	Eye Squint Left
4	Throw Weapon	R	Quaternary	Upper Right	Moderate	Eye Squint Right
5	Sheikah Slate	Minus -	Tertiary	Solo	High	Head Pitch Left
6	Pause Menu	Plus +	Tertiary	Solo	High	Head Pitch Right
7	Move	Left Stick (NB)	Primary	Lower Left	High	Directional Lips Push
8	Interact	A	Secondary	Lower Right	High	Right Cheek Puff
8	Cancel	B	Secondary	Lower Right	High	Right Cheek Suck
8	Jump	X	Secondary	Lower Right	High	Left Cheek Suck
8	Attack	Y	Primary	Lower Right	High	Left Cheek Puff
9	Inventory Selection / Whistle	Directional Pad	Quaternary	Lower Left	Low	Say "Up," "Down," "Left," "Right"
10	Move Camera	Right Stick (NB)	Primary	Lower Right	High	Head Roll, Head Yaw

The following is in regard to the buttons labeled on the back of the controller.

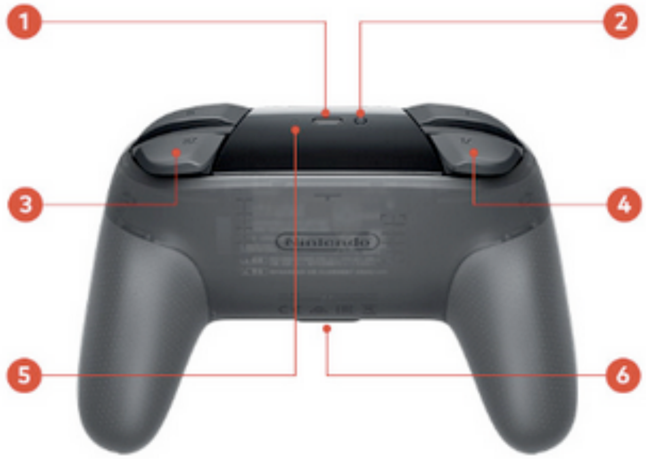


Figure 20: Nintendo Switch Pro Controller Map, Back

Only labels 3 and 4 on Figure 20 are buttons --- the rest are labeling controller ports and other information.

Table 3: *Breath of the Wild* Hands-Free Controls Example (Back)

#	*Action*	*Button*	*Frequency*	*Quadrant*	*Gesture Comfort*	*Gesture*
3	Use Bow	ZR	Tertiary	Upper Right	Moderate	Wink Right
4	Raise Shield	ZL	Tertiary	Upper Left	Moderate	Wink Left

This controller map was created as an example of how a player may pair their most comfortable gestures with the most common actions used in gameplay. If possible, it is helpful to find a few gestures that could be done simultaneously and assign each to a quadrant of the controller. This allows the player to access the full range of button combinations.

5. Technical Project: Control Recommendation Demo

What if you decided not to use your hands for gaming or using a computer -- what would be your first attempt at using your device hands-free? I asked myself this question when approaching the technical aspect of my project. Would I try voice control? Face-tracking? Eye-tracking? As explained in Chapter 4, there are numerous options available, each with unique advantages and challenges. However, the technology is not the only unknown variable -- I am unknown, too. Could I play *Pong* with my eyebrows? What about *Pac-Man* with my lips? Could I type with an eye-tracker? I have no idea how dexterous my face is. I do not expect that anyone knows the accuracy by which they can use the muscles in their face.

This chapter describes the next step in my streamlined hands-free gaming experience – the assistant to games categorized by frequency of use. Here I will describe how people may learn which hands-free methods work the best for them.

5.1 Project Origins

My technical project began as an exploration of hands-free gaming. I wondered how I could create a hands-free game, perhaps using a virtual reality headset. As seen in Chapter 3, there are already some virtual reality games that don't require hand controls and only ask the player to use head motion to play. The Unreal Engine[1] makes creating

[1] The Unreal Engine is a game engine created by Epic Games. It is used to produce video games as well as architectural and automotive visualizations, movies, TV shows, live events, training simulations, and more.

virtual reality games simple with its out-of-the-box VR template. I had some experience building games in the Unreal Engine and felt that building a hands-free game would be an opportunity to learn more about hands-free game design.

What bothered me about this project idea was that it seemed disconnected from the point of my thesis – that approaching game control maps from a perspective of frequency of use could help make video game controls more adaptable and games more accessible. While I could certainly learn from the aforementioned project, the overall concept felt peripheral to my main points. This process repeated with a few other ideas—creating games that could be played using face-trackers or eye-trackers, building a VR simulation, creating a game that could be played in virtual reality and on a 2D platform, etc. I surfed internet forums discussing the best video games and devices for people with disabilities and was inspired by the dozens of home-made hacks people had made for their own hands-free gaming purposes. I tried out some free online software designed to give users hands-free control over their mouse, including an eye-tracker designed to replace a computer mouse named GazePointer.[2] Eventually, I realized that I would not be happy with any idea that did not contribute directly to the ideas presented in my thesis. I needed something with a more explicit purpose.

[2] Link to view and download GazePointer: https://sourceforge.net/projects/gazepointer/

While playing with the eye-tracking software GazePointer, it occurred to me that both the tracker and my ability to cooperate with it were vital to the overall success of using the tracker as a mouse. Some of my family members had an easier time working with the eye-tracker, while others only got a headache. Even though all of us have virtually the same abilities with our eyes, our abilities in conjunction with the eye-tracker varied greatly. Further, it was difficult for any of us to determine who may achieve greater accuracy with the eye-tracker than anyone else.

This led me to consider project ideas that could help users learn about themselves as well as their software or technology. For the best user experience playing hands-free video games, there must be a strong sense of control and comfort between the user and the device, whether that be an eye-tracker or something else.

5.2 Project Goals

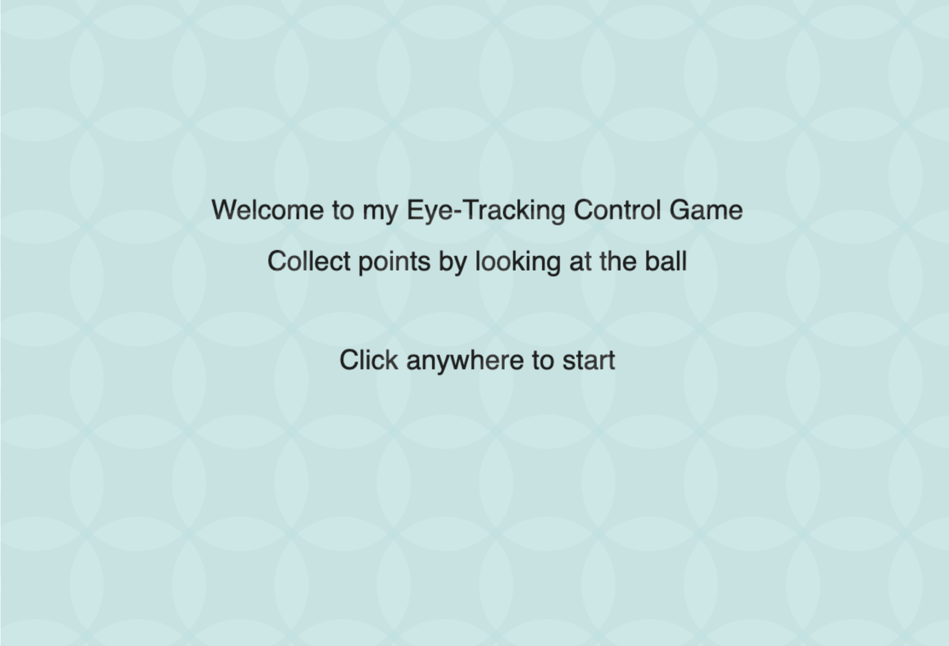
Let's say I wanted to play *The Legend of Zelda: Breath of the Wild* hands-free. I determine which controls in the game are primary, secondary, tertiary, and quaternary. How do I pick the method by which I will operate the primary controls? Without at least a moderate understanding of which gestures are the most comfortable for me, the best I could do is guess. Trial and error is a fine way to learn if I could control Link with my eyebrows or lips, but this process could be made faster and less frustrating if I knew in advance which gestures I could complete with the most comfort and accuracy.

I propose that a game could help people determine which gestures may be the best for each control. This game would be designed to help anyone learn about their face or head dexterity, not just people seeking hands-free gaming technology. This project is inspired by universal design principles in that it seeks to be a useful tool for any curious person, regardless of ability or reason.

My goal in this project is to create a demo of what one of these games may look like. Perhaps more importantly, my goal is not to create a game that is scientifically proven to help people determine their optimal primary control method. That is outside the scope of this thesis, although admittedly would make an interesting Ph.D. project. I wish to demonstrate what could, with more time and technical knowledge, be created for the future of adaptive gaming controls and hands-free technology.

5.3 Project Description

I built a simple game using p5.js, which is a JavaScript library focused on making coding more accessible for people working on creative projects. The game is designed to be played using an eye-tracker but can be played in any way where the user can control a computer mouse.



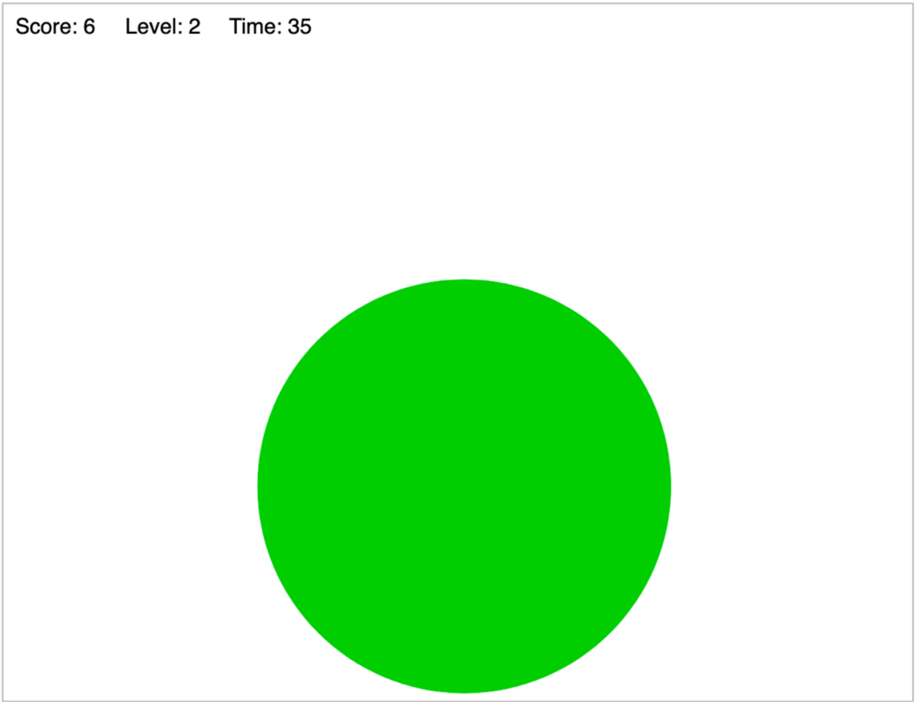
Welcome to my Eye-Tracking Control Game

Collect points by looking at the ball

Click anywhere to start

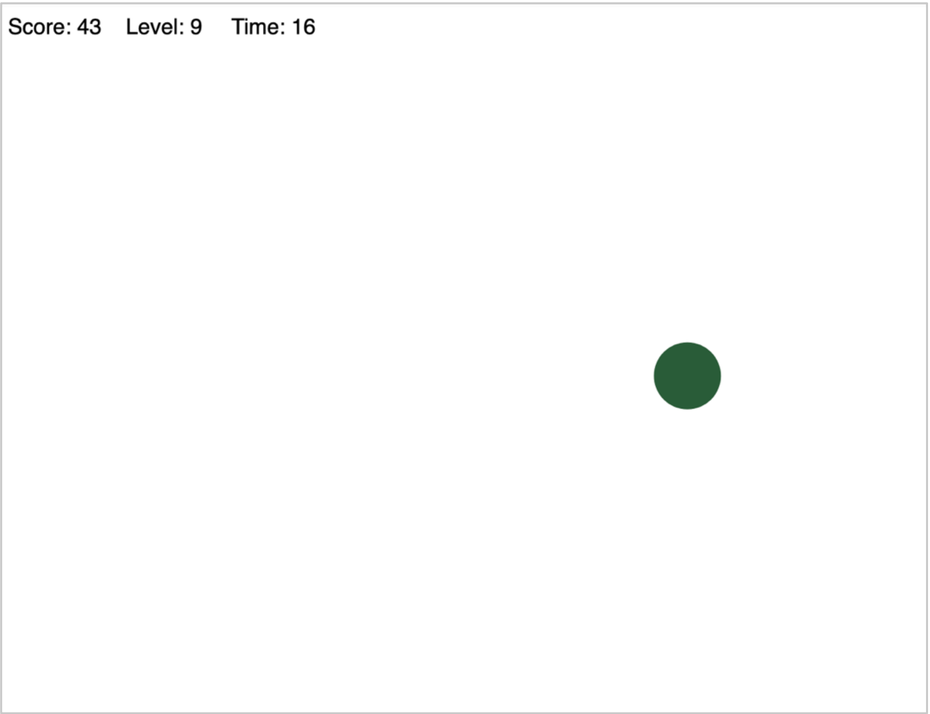
Figure 21: Start screen of my eye-tracking game

The game presents the player with circles of an ever-decreasing diameter that change position and color every two seconds. If the player touches the circle with the mouse, the circle reappears in a new location and the player scores a point. The circle decreases in diameter for every five points that the player scores. The player has one minute to score the highest score possible.



Score: 6 Level: 2 Time: 35

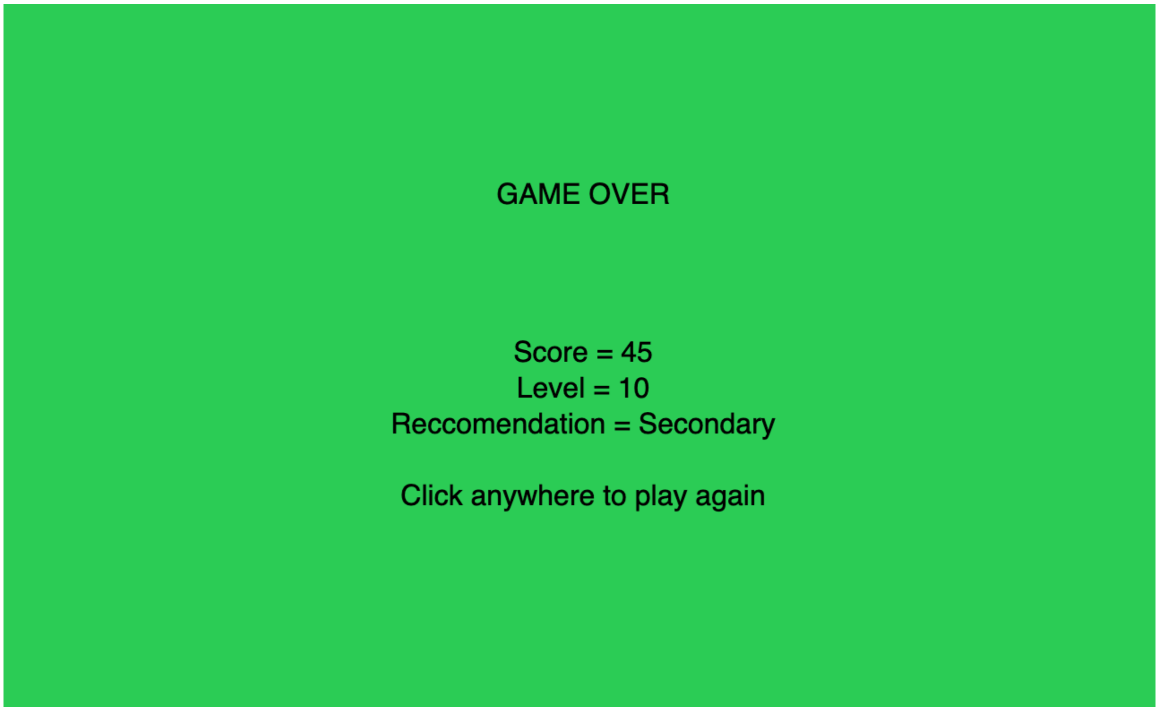
Figure 22: Eye-tracking game gameplay, Level 2



Score: 43 Level: 9 Time: 16

Figure 23: Eye-tracking game gameplay, Level 9

At the end of the game, a screen showing the player's score is displayed. This screen also offers a recommendation for the player based on the smallest circle diameter the user could touch. The recommendation is if the player's mouse control method is suitable to be a primary, secondary, tertiary or quaternary control method. For example, if the player was using an eye-tracker to control the mouse and the smallest diameter reached was larger than the size of a quarter, then this person would likely struggle to accomplish tasks such as typing or selecting small icons with the eye-tracker. This would not make a good primary control system. Perhaps eye-tracking could be used for something that requires less accuracy, such as selecting a direction to move when navigating on Google Maps.


GAME OVER

Score = 45
Level = 10
Reccomendation = Secondary

Click anywhere to play again

Figure 24: Eye-tracking game end screen

With more time, this project could be enhanced to have a more intelligent recommendation system than what currently exists. For the purposes of this thesis, the game I created is only a demo to allow users to imagine what could be created with more time and technical skills. A link to the live P5.js demo can be found in the footnotes of this page.[3]

5.4 This Project's Context in My Larger Accessibility Concept

While such a game as I described could be a useful assistant to a gamer seeking accessibility, this project is ultimately of limited impact without much more data informing the control recommendations. Control recommendations go beyond user comfort with specific gestures – some controls are binary, while others are not. Some controls need to be pressed in sequence, simultaneously, or held for periods of time. Optimized recommendations need information about the user's hardware, available space, and the games they want to play. All of this and much more would be required for the most user-centric control recommendations experience.

After much deliberation, I decided to take everything I have learned about game accessibility and outline a software that goes beyond this technical project and my programming skills. This interface software concept is designed to combine the efforts and information from game designers, hardware designers, and users to create a

[3] https://editor.p5js.org/andybrucculeri/present/et_AbASON

comprehensive game controls recommendation API and GUI (graphical user interface). In this software, a game that helps a user determine which gestures are the most comfortable may be just a small part. The game as described in this chapter would be a way of collecting data from the user, which is only one of the three branches of data necessary for the API software concept.

Because this concept remains ultimately separate from my technical project, I have decided to describe it at length in a separate chapter, which directly follows this one.

6. A Interface Software Concept

The concepts discussed throughout this thesis may be used to create a software product or suite. The interface software may have one or more automated interfaces to provide accessibility features for gaming. The interfaces are used to facilitate communication between the interface software's three general data sources – game software or game developers, hardware or hardware developers, and game users. This arrangement is modeled in Figure 25, where the three data sources are shown interacting with the interface software through APIs or other interfaces. The interface software may deliver at least the following functionality to make it easier to provide accessibility in gaming:

1. Match user-specific capabilities to games using empirical and survey data from the user, games, and game developers
2. Help a user play a specific game by matching user-specific capabilities with a particular game over a spectrum of hardware options
3. Suggest games to a user based upon an analysis matching user-specific capabilities with empirical or survey data regarding a spectrum of game
4. Suggest hardware to a user based upon matching empirical and survey data

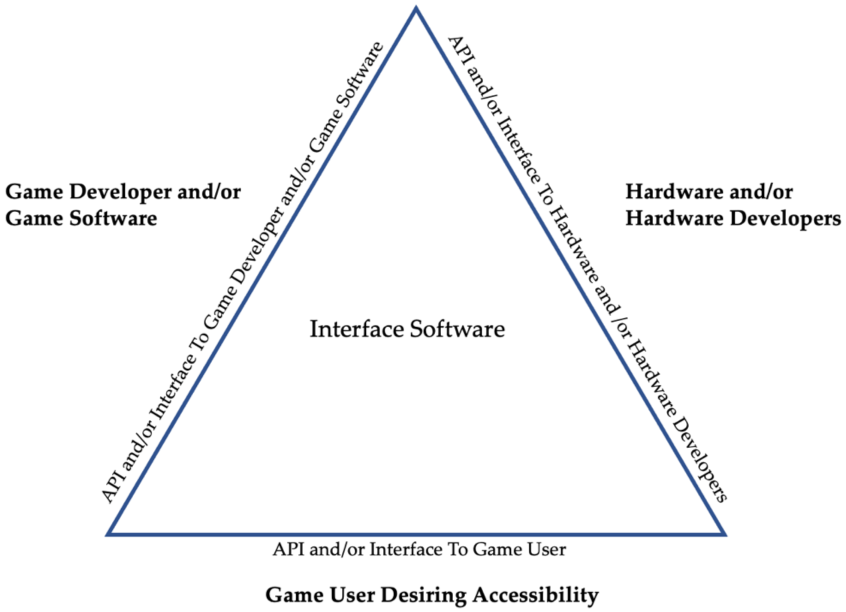

Game Developer and/or
Game Software
Hardware and/or
Hardware Developers
API and/or Interface To Game Developer and/or Game Software
API and/or Interface To Hardware and /or Hardware Developers
Interface Software
API and/or Interface To Game User
Game User Desiring Accessibility

Figure 25: The three data sources interacting with the interface software

6.1 Game Developers and Game Software

Game developers and game software provide important data to the interface software because relevant and reliable information about each game is required to achieve the software's objectives. Game-related information can be analyzed with respect to user capabilities to form functionally-optimized correlations between users, games and hardware. As discussed at length in this thesis, one primary goal is to optimize gameplay for gamers who cannot effectively use traditional hand controllers. Hand controllers have over a dozen buttons, as shown in the Xbox controller illustrated in Figure 26. In order to suggest the functionally-optimized correlations listed above, the interface software requires information regarding the nature and importance of each control on each controller. Game developers are already likely in possession of the required usage statistics. For example, usage statistics can empirically illustrate frequency, duration and range statistics (for non-binary controls) relative to each particular control. Similar information may also be obtained through questions, interviews or surveys with the game developer or even experienced game users.

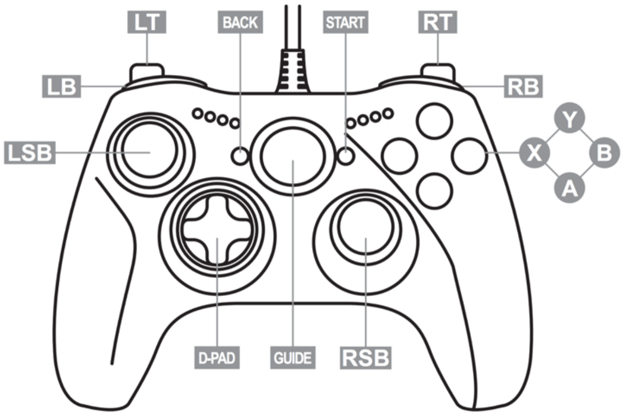


Figure 26: Illustration of an Xbox controller

One way to obtain developer information is through an interface directed to developers or others sufficiently knowledgeable about the game. For example, a simple GUI may be used to solicit answers to questions about gameplay. The GUI may request answers through a Q&A implemented with checkboxes or pull-down menus. For each mode of a game, the interface may request answers regarding each game control. The questions may apply to each applicable control on a keyboard or traditional controller, including both binary controls (such as buttons) and multi-dimensional controls (such as analog sticks). The following are examples questions for binary controls:

- whether the control is required for gameplay

- ranking of the control either on a standard scale (such as 1 – 10) or among the other binary controls used by the game
- anticipated frequency the control will be used during gameplay (such as constantly, often, periodically, sparsely)
- how commonly the control is held for more than a few seconds
- how often the control is held for more than five seconds
- how often the control is used in combination with other controls
- which other controls are used in combination with the control under inquiry, and for each, how often
- which controls are used in close sequence to each other and how often
- which controls are used in rapid succession and how often

The following table shows an example of a graphical interface for accepting data such as the control characterizations discussed above, with LB and LT as the example buttons.

Table 4: Example of GUI asking for data on buttons

	LB	***LT***
Importance of control to game?	Select ranking or grade	Select ranking or grade
Frequency of control use in game?	Select ranking or grade	Select ranking or grade
Common duration of active state in game?	Select ranking or grade	Select ranking or grade
Used in combination with other controls?	Select yes or no	Select yes or no
Select other control 1	Select a control	Select a control
Frequency of use with other control 1?	Select ranking or grade	Select ranking or grade
Used in close sequence with other controls?	Select yes or no	Select yes or no

As shown in Table 4, questions may be aligned with each control. While only LB and LT are shown, these questions may be asked about any or all controls on all controllers the game natively employs. Table 4 also shows queries that require nesting to gather all necessary information. For example, in gathering information regarding using a combination of controls, in order to use a GUI in table format with pull-down or menu selections, significant nesting would be necessary as shown – at least two nested questions for each potential combined control.

In another example, a GUI or an API may be fashioned to obtain empirical data regarding gameplay by one or more users. This type of data may be received in any usable form, such as flat files (spreadsheet form or comma-separated) or database files. The GUI or API may facilitate the interface software to analyze the data patterns and provide empirically-guided assessments of the nature and common use of each traditional control. For example, data for a particular game will reveal the frequency of use for each control, the frequency of combined control use, and the common control transitions (or which controls follow rapidly after a reference specific control), among other things. An API may be used to map the data fields with the corresponding query areas, such as shown below.

(Game;Game Mode;Controller;[Column definition fields];[Row definition fields]),

Or

(Game;Game Mode;Controller;[Column definition field X];[Corresponding row definition field X])

Figure 27: Sample API formats

The following are definitions for the sample API formats:

Game: The "Game" is information identifying the game for which data is submitted. The game identification may be selectable as part of the API or it may be included as an open-input text field.

Game Mode: The "Game Mode" refers to a particular mode in a game that has multiple modes. For example, in the 2020 implementation of *Fortnite* by Epic Games, the common mode of play is "Battle Royale," although multiple other modes are available for players to choose from (including "Save The World," "Creative," and "Party Royale"). The Game Mode field in the API identifies the mode because different controls may have different functions or importance in different modes. Like the game field, the API may provide the game mode as a selectable code, an option from a predetermined list, or an open-input text field. The Game Mode may refer to modes of a game at any level of abstraction. For example, using the *Fortnite* example, “Battle Royale” may be used as mode, or at a lower level of abstraction, the mode may be limited to “Battle Royale” from a specific season (or iteration of the game from a certain date).

Controller: The “Controller” field identifies the controller model used in the data to be submitted. The model may be based upon specific controllers available for selection through the API. For example, the Xbox controller model would indicate that the controls described in the data are those used on an Xbox controller. Of course, other controllers may also be represented and a generic controller model may be included in the selection. The API may define the controls for the generic controller, preferably in a way that aggregates the controls from multiple controllers.

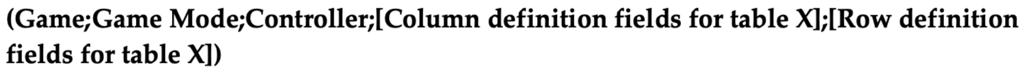
[Column definition fields] and [Row definition fields]: By referring to column and row fields, this example of the API appears limited to flat data file forms, but it is not. The example also shows that the API does not force one type of data to be carried in either rows or columns. Different implementations of the API merely require correlation between elements data (such as game data) and operative rows or columns. In one version of the API, the developer may use the API to correlate columns with the controls applicable to the game and the identified controller. For example, assuming columns are used to carry the controls, each potential control (LB, LZ, RB, RZ, etc.) occupies a column. Further, if columns are used for controls, then rows may be used to specify corresponding data acceptable by the API such as:

- Average or median control interactions per game
- Standard deviation for control interactions per game
- Average or median control interactions per minute
- Standard deviation for interactions per minute
- Average or median duration of activation
- Standard deviation for duration of activation
- Average or median duration between activations of the control
- Standard deviation for duration between activations of the control
- Average or median instances of combined use with other controls per game or per minute
- Maximum number of other controls in combined use
- Identity of other controls in combined use
- For each other control (may be a row for each control), average or median time of combined use per game
- Standard deviation for each other control, average or median time of combined use per game
- For each other control, average or median duration of combined use per minute

- Standard deviation for each other control, average or median duration of combined use per minute.

A sample version of the API may express [Column definition fields] as an ordered and structured list of the controls such as [LB, LZ, RB, RZ . . .], although other symbolism may be used for each control. Similarly, [Row definition fields] may be expressed using structured and sequenced documented mnemonics for each acceptable data type, such as the data types listed above.

The game data from a database may be used by the interface software. For example, each table from the database may be used as a flat data representation (such as row and column correlation to data types and controls). Alternatively, the interface software may interact with a database like any other application -- through the database management system (DBMS). However, even with access through the DBMS, the interface software may require an understanding of the meaning of each data field. That meaning can be sufficiently conveyed by correlating the database's data fields with the applicable game controls. Using the discussion above as illustrative, each game control may be correlated with multiple data types (or database fields) such as those listed above. Sample API calls for making these correlations might appear as follows:



(Game;Game Mode;Controller;[Column definition fields for table X];[Row definition fields for table X])

Figure 28: Sample API calls for making correlations

The above sample API call allows for defining columns and rows for multiple tables of a database.

(Game;Game Mode;Controller;[DBMS Correlation])

Figure 29: Standard or structured sample API call

The above sample shows that database management system information may be conveyed to the interface software through a standard or structured call. Most databases and development languages have well-established structured systems (such as APIs) for use of the database from a certain software environment (for example, JDBC – java database connectivity).

The following sample shows how controls might be correlated with database fields.

(Game;Game Mode;Controller;[Controls list][Corresponding field list]),

Or

(Game;Game Mode;Controller;[Control X][Corresponding field X]),

Figure 30: Sample API correlated with database fields

Another way to receive information about a game is for the game developer to provide an API for its game usage data, preferably through the internet. For example, the game developer may provide APIs that allow an authorized third party, such as the interface software, to obtain usage statistics for the game's controls. The API can be used by the designers of the interface software to ascertain the usage data necessary to make optimized correlations with a set of accessibility controls, such as controls of a hands-free headset.

While it may seem burdensome for every game developer to provide these types of APIs, many games could provide APIs of this type merely by making adaptations to the most common game development engines. These engines are used to develop and compile an extremely large number of commercially available games. For example, development systems such as Unity or the Unreal Engine provide platforms to develop and implement games for commercial exploitation. By definition, these development platforms map game action to traditional game controls such as those in handheld controllers. Thus, the production of game usage data or an API to provide access to such data may be built into the development platform and then applied to every game produced on the platform. Again, this may sound onerous but much if not all of this work is already done. For example, both Unity[1] and the Unreal Engine[2] provide facilities such as APIs for analytics. These facilities might be used as-is or adapted for purposes of the interface software. Therefore, the interface software may be able to obtain information regarding a huge number of games by accessing APIs or other information sources that are provided by the game developers.

[1] Unity: https://docs.unity3d.com/Manual/UnityAnalyticsRawDataExport.html

[2] The Unreal Engine: https://docs.unrealengine.com/en-US/Gameplay/Analytics/index.html

6.2 Hardware and Hardware Developers

Many types of hardware may couple to the interface software. For example, all of the following and more might be useful: cameras, all types of motion sensors (such as accelerometers, infrared sensors and LIDARs), proximity sensors, and any device that can detect sound, movement, light or environmental changes of any kind. In addition, modern sensors such as cameras may operate with dedicated software to analyze the sensor output and provide conclusions that might be used by the interface software. The output of this type of software is treated as hardware for the purposes of this discussion. Some examples of these types of hardware/software pairings are: camera output analysis providing conclusions about the status of eyes (open or shut), mouth (open or shut, smiling or frowning), or other facial status; or, LIDAR output analysis providing a conclusion regarding the distance between two objects.

For each hardware device, the interface software may provide an interface such as an API so that a device can be interpreted by the interface software. For example, the API will allow a developer to pair the device with the interface software so that the interface software may in turn provide the functionality of the device for gaming accessibility. The output of a device may be used alone or in combination with other device outputs or game data to map to accessibility game controls. For example, a motion sensor may output a range of values that indicate a head nod to be analyzed by the interface software. The head nod may then be mapped to one or more binary controls. A single binary control may correlate the sensor output to either a nod up or a nod down. Alternatively, multiple binary controls may correlate the sensor output with multiple sub-ranges in the full range of motion indicated by the device output.

The following is an example of a device with an output range between 0 and 19 being used for a single binary control:

0 through 9 equals binary "on"

10 through 19 equals binary "off"

The next example demonstrates how a single binary control may have three or more states:

0 through 6 equals binary "on"

7 though 12 equals no decision

8 through 19 equals binary "off"

The following is an example of a device with an output range between 0 and 19 being used for two binary controls:

0 through 4 equals binary "on" for binary control A

5 through 9 equals binary "off" for binary control A

10 through 14 equals binary "on" for binary control B

15 through 19 equals binary "off" for binary control B

The following is an example of a device with an output range between 0 and 19 being used for two binary controls with three or more states:

0 through 3 equals binary “on” for binary control A

4 equals no decision

5 through 8 equals binary “off” for binary control A

9 through 10 equals no decision

11 through 14 equals binary “on” for binary control B

15 equals no decision

16 through 19 equals binary “off” for binary control B

As another example, a combination of motion sensors may output multiple ranges of values that indicate a two-dimensional position of the user’s head. That two-dimensional position may in turn be analyzed by the interface software and mapped to game controls. The two-dimensional head position may be mapped to a two-dimensional game control such as an analog stick. One potential way to make such a mapping is to correlate ranges of (X,Y) sensor data with ranges of (X,Y) analog stick data.

Since discrete sensor hardware devices tend to have standardized output, a correlating standard API may be designed for a large group of common devices, such as hardware IR, Lidar, accelerometer, camera, etc. However, the sensor output alone does not provide any information regarding the state of a user because any particular sensor may be applied to a user in any number of ways. Therefore, the API should provide for correlation of the sensor output with a user status.

```
([sensor output range X],[corresponding user status X])
```

Figure 31: Sample API correlating the sensor output with a user status

The use of the variable “X” in Figure 31 is to indicate that this type of API statement generically provides for correlating multiple output ranges of the sensor with multiple user status situations.

As discussed above, some examples of user status are as follows:

- Right eye open / squint / closed
- Left eye open / squint / closed
- Gaze angle of right eye
- Gaze angle of left eye
- Right eyebrow raised / relaxed
- Left eyebrow raised / relaxed
- Right cheek inflated / relaxed
- Left cheek inflated / relaxed
- Smile present
- Frown presence
- Top lip recessed / prominent
- Bottom lip recessed / prominent
- Jaw position (3 dimensions)

As noted in section 3.3, no discussion of hardware is complete without recognizing the current status of virtual reality hardware, which may provide all the sensor activity necessary for an early version of the interface software. Since virtual reality hardware already evaluates the use of its sensors for gaming and provides access to gaming software developers, the interface software may make use of the same access to provide its functionality. Further, game development platforms (such as Unity or the Unreal Engine) are already programmed to interact with a variety of hardware and provide those interaction abilities to games developed on the platforms. In other words, game development platforms already interpret hardware outputs and correlate those outputs with user status so that game software may react in gameplay. With respect to games developed on these platforms, the interface software can simply leverage the hardware interpretations provided by the development platform software. This is why the most efficient manner to add new hardware capabilities to scores of games may be to enhance the game development platforms to serve a range of accessibility hardware by correlating the hardware output with the desired user status, such as those discussed above.

6.3 Gamers Desiring Accessibility

A primary purpose of the interface software is to serve the particular needs and desires of game users requiring accessibility. Those needs and desires may be highly individualized to the user. The interface software may determine those needs and desires through two primary techniques. First, a GUI may be used to query the user or their caretaker regarding the user's desires and abilities. The following provides a sample of information that may be obtained from a user through a GUI ,such as a menu with pull-downs, checkboxes, or free-form fields for user input:

- Identifying types or genres of games desired (first-person shooter, adventure, casual, etc.)
- Identifying a favorite game
- Identifying a list of favored games, perhaps ranked
- Identifying hardware that the user has or may acquire
- Queries regarding user abilities relating to fingers, hands, toes, feet, arms, elbows, shoulders, wrists, knees, ankles, neck, mouth, lips, jaw, cheeks, tongue, eyebrows or eyes
- Querying regarding user proficiency with different identified hardware devices

In addition to a GUI, the interface software may interpret sample gameplay or playtesting by the user to determine the user proficiency and skills (such as listed and discussed above) or user preferences. Gameplay may be analyzed in a number of ways. The easiest way to analyze gameplay may be through game system resources or game development kit resources provided for developers. Depending upon the game system or development kit, these resources may include activity logs, gameplay statistics and other resources. The data received from these resources may be analyzed by the interface software to determine the relative abilities and preferences of the user.

Another way to analyze gameplay is to run a software module of the interface software on a game system while a user is playing enabled by accessibility hardware such as a headset. The module may be designed to collect and log any data desired so as to monitor any user activity that the sensors are capable of perceiving and correlating that information with gameplay. In the absence of special arrangements, the module may require the user to play in a system developer mode so that the game and the interface software module may be simultaneously run.

Yet another way to analyze gameplay is through playtesting modules that may be a part of the interface software. An illustrative playtesting software may be a game or an exercise, such as the eye-tracking game discussed above, that tests a user's abilities and preferences when using accessibility hardware. This concept is discussed at length and demoed in Chapter 5 as the technical aspect for this thesis. Playtesting modules have the advantage of directly accessing the user skills desired by the accessibility designers. Of course, by itself, a playtesting module would not necessarily collect or log the appropriate data, so the techniques discussed above remain relevant.

Finally, user analysis and interaction may be materially aided by using or altering the facilities provided by game development systems such as Unity or the Unreal Engine. These game development systems may equip game software with the ability to run a monitoring module for collecting accessibility data or to run playtesting modules. Importantly, because the game development system is inextricably part of the games themselves, a data collection module or playtesting module may be extremely tailored to the play or a particular game with respect to particular accessibility hardware employed by a particular user.

6.4 Using the Interface Software to Provide Accessibility

In order to make accessibility suggestions as discussed above, the interface software may use the data collected through the interfaces to make inferences regarding the correlation of hardware or traditional controls with accessibility controls. The interface software may present the collected data to the user in a visually accessible form and allow the user to make the control pairings. A more streamlined process requires that the interface software make the final control correlation determinations rather than the user.

Figure 32 shows that a model may be used to aid with making final determinations regarding control correlations. A purpose of the model is to determine and use the relevant data in the priority order suggested by the collected data. Designers may differ on the exact nuances of such a model, and the determination of the best or better models will undoubtedly be a matter of trial and effort. Further, differing models may be more useful for differing types of users or games. For these reasons, the interface software concept is envisioned to accommodate any model.

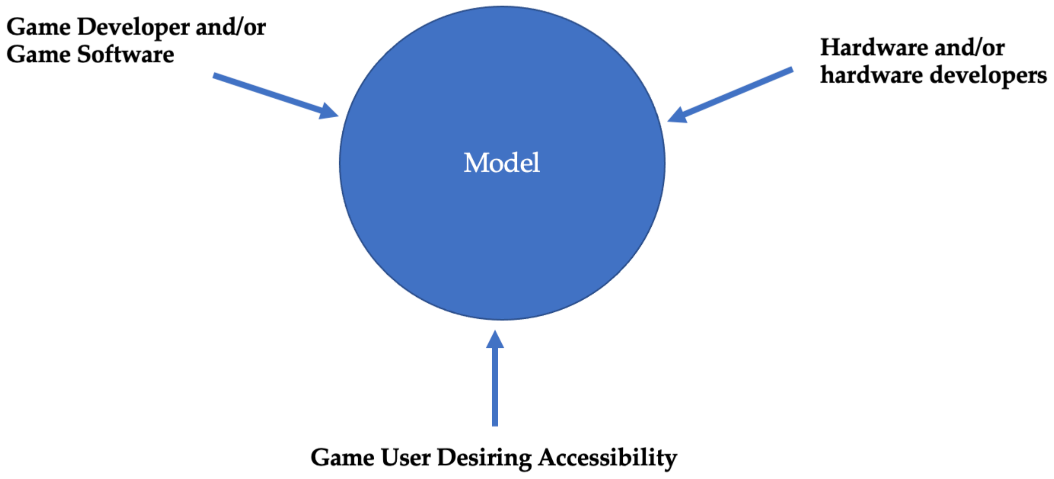


Figure 32: Illustration of model accepting data sources

One example of a model is illustrated in Figure 33, which shows that the illustrative model seeks to use the collected data to correlate controls. Figure 33 shows on the left an exemplary list of accessibility controls, and on the right side an exemplary list of hardware and traditional controls.

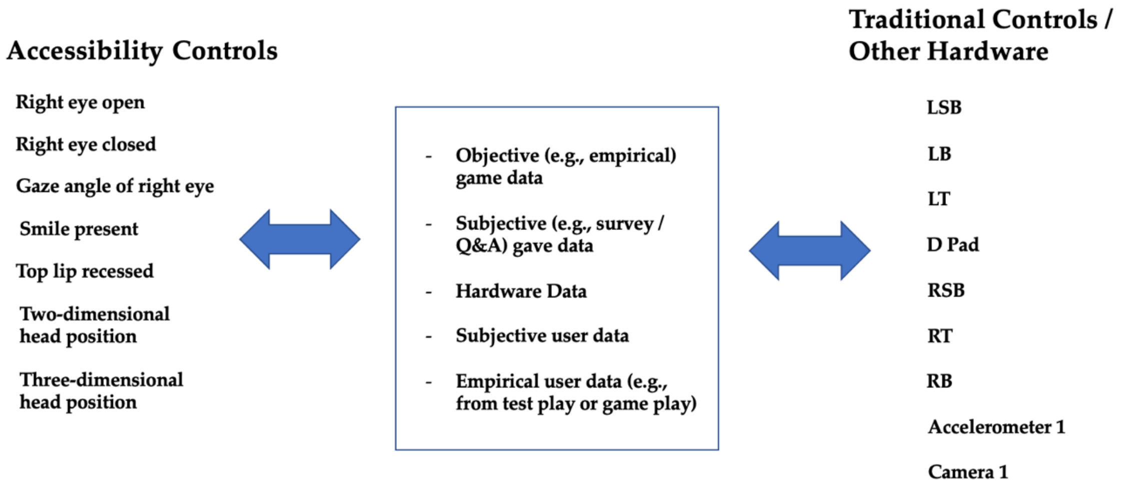

Accessibility Controls
Right eye open
Right eye closed
Gaze angle of right eye
Smile present
Top lip recessed
Two-dimensional head position
Three-dimensional head position
- Objective (e.g., empirical) game data
- Subjective (e.g., survey / Q&A) gave data
- Hardware Data
- Subjective user data
- Empirical user data (e.g., from test play or game play)
Traditional Controls / Other Hardware
LSB
LB
LT
D Pad
RSB
RT
RB
Accelerometer 1
Camera 1

Figure 33: How the interface software uses the collected data to correlate controls

Figure 33 also shows an exemplary list of the data collected by the interface software and discussed above. The concept of this illustrative model is built around the controls, which the model views as objective. There is a finite and known set of controls or hardware that may be accessed by a particular game and there is also a finite and known set of accessibility controls available to a user and the software. The concept of the model is to use the collected data to create appropriate and ideal correlations between the two types of controls.

Figure 34 shows a process that may be implemented to apply the illustrative model. The process may be implemented in software run on any suitable computing hardware.

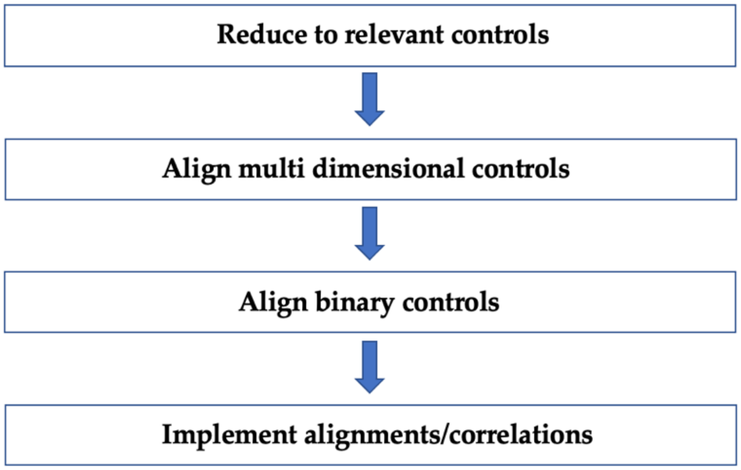

Reduce to relevant controls
Align multi dimensional controls
Align binary controls
Implement alignments/correlations

Figure 34: High-level apply model flow chart

Initially, the process seeks to reduce the set of traditional controls to the relevant set under consideration, as shown in Figure 35. Therefore, any controls that are unused by or unavailable to the game should be eliminated, reducing the list to only practically relevant controls. In addition, as shown in Figure 35, during this first step, the list of accessibility controls should also be tailored to the practically available set.

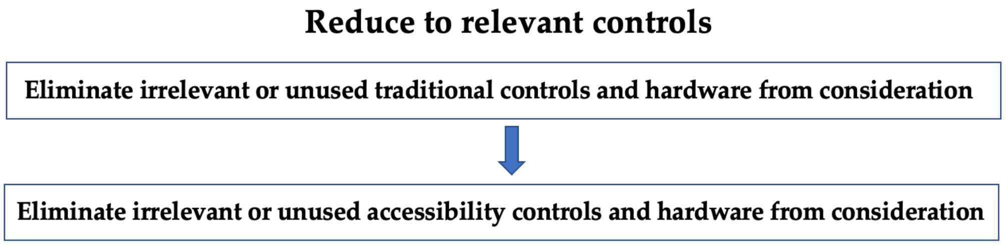

Reduce to relevant controls
Eliminate irrelevant or unused traditional controls and hardware from consideration
Eliminate irrelevant or unused accessibility controls and hardware from consideration

Figure 35: Reduce to relevant controls nested flow chart

For example, the list might not include accessibility controls that are unavailable to the user due to hardware constraints or due to a user's known physical limitations. The second step in the process is to use the collected data to align multi-dimensional traditional controls with accessibility controls of like dimension. The third step of the process similarly aligns binary controls from the traditional list with the accessibility list. While opinions may differ on priorities for alignment, in one possible model implementation, the following list of priorities is applied:

1. User mandates regarding the use of a control
2. Align traditional multi-dimensional controls requiring a certain degree of accuracy with multi-dimensional accessibility controls for which the user has a predicted accuracy of a similar degree or higher.
3. Align traditional multi-dimensional controls requiring a certain level of repetitive use with multi-dimensional accessibility controls for which the user has predicted tolerance for a similar level of repetitive use or higher.

4. Align traditional multi-dimensional controls requiring a certain level of rapid directional change with multi-dimensional accessibility controls for which the user has a predicted ability of rapid directional changes at a similar level or higher.
5. Align traditional binary controls requiring a certain level of repetitive use with binary accessibility controls for which the user has predicted tolerance for a similar level of repetitive use or higher; but only if: the alignment allows for combined controls uses that are predicted as tolerable for the user; or there is no tolerable alternative.
    a. In aligning controls with respect to combined control usage, binary controls that are predicted best for combined uses should be prioritized as long as the required level of repetitive use is not intolerable to the user based upon data or user mandate.

The final step of the process illustrated in Figure 34 calls for implementing the alignments, which refers to changing the state of the gaming systems so that the preferences are implemented. Simply put, this step involves either programmatically or manually setting the game to use the selected controls for the identified purposes. By way of example, many games allow controls to be mapped per the user's direction. This step completes that mapping, although perhaps with a set of controls that is not readily available at the time of this writing.

Figure 36 shows further detail to the step of aligning multi-dimensional controls. As shown, the first step is to determine the set of multi-dimensional controls required for playing the game under consideration. This may include creating a list of multi-dimensional controls and noting the requirements use indicators for each. Next, a determination is made regarding the available multi-dimensional accessibility controls. Similarly, a list may be constructed with each control accompanied by its constraints and usability data. Next, user mandates or learning from prior attempts should be applied. For example, if a user has mandated a certain alignment or declined to use a certain accessibility control, the user's indication should be followed. As another example, if a prior attempt of the same analysis has proven undesirable to the user, then any indication from that experience should be applied (for example, a user having reaction or endurance issues with one accessibility control may indicate using a different accessibility control). In the final step, the alignment is made and priorities should be applied such as the priorities listed above.

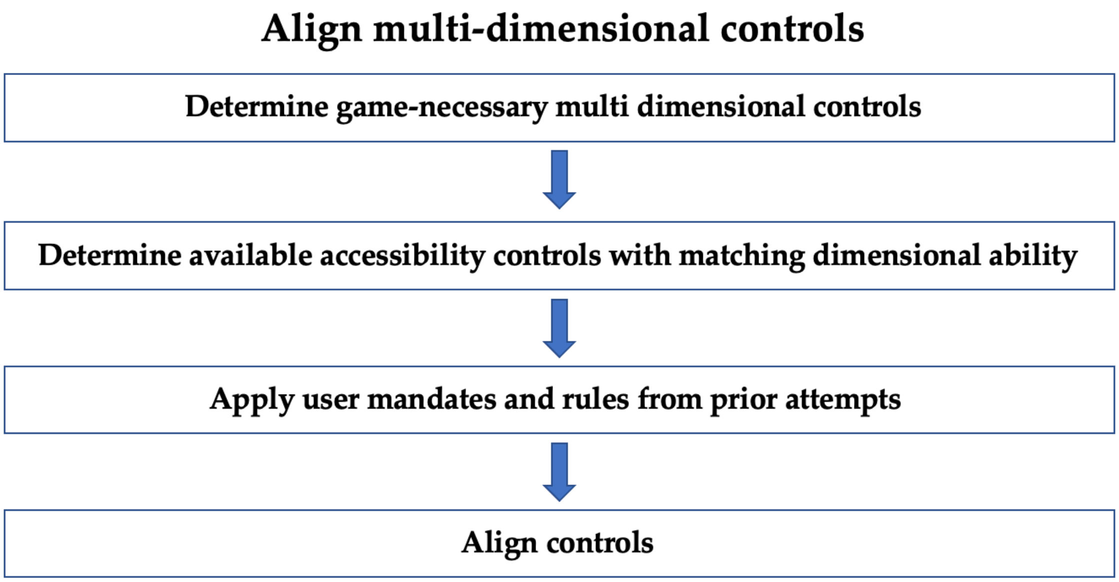

Align multi-dimensional controls
Determine game-necessary multi dimensional controls
Determine available accessibility controls with matching dimensional ability
Apply user mandates and rules from prior attempts
Align controls

Figure 36: Align multi-dimensional controls nested flow chart

Figure 37 shows further detail on the step of aligning binary controls. The discussion with respect to Figure 36 applies here as well.

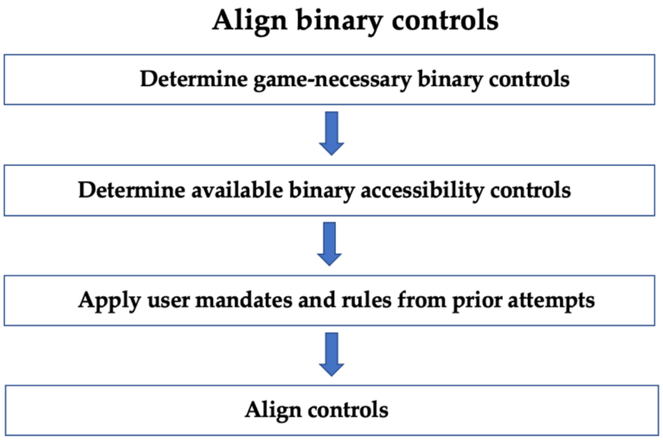

Align binary controls
Determine game-necessary binary controls
Determine available binary accessibility controls
Apply user mandates and rules from prior attempts
Align controls

Figure 37: Align binary controls nested flow chart

6.5 Concluding Suggestions to Enhance Accessibility

In the beginning of this chapter, four ways to enhance user accessibility were suggested. Each may be implemented using the model and illustrations provided above.

a. Match user-specific capabilities to games using empirical and survey data from both the specific user and the games and game developers.

The model and process discussed above are directed to this generic use of the interface software.

b. Help a user play a specific game by matching user-specific capabilities with a particular game over a spectrum of hardware options.

In adapting the processes discussed above, if a user wishes to play a specific game, the hardware controls should be limited to those applicable to that particular game.

c. Suggest games to a user based upon an analysis matching user-specific capabilities with empirical or survey data regarding a spectrum of games.

To adapt the model and processes discussed above, each game under consideration should be processed separately and a compatibility score should be applied to each. The compatibility score may be based on how often or by how much the predicted user abilities for each accessibility control exceed the predicted requirement for the traditional control. The highest scored games may be recommended to the user.

d. Suggest hardware to a user based upon matching empirical and survey data about the user.

To adapt the model and processes discussed above, the process should be run multiple times using:

1. A core set of accessibility controls or a large set of generic accessibility controls should be used in several runs of the process. Each run of the process should use a different set of traditional controls, and each set should be scored as discussed above. Each set of traditional controls should be based upon a single existing hardware system, such as different VR headsets or different combinations of discrete hardware and VR headsets.
2. The user may be advised of the hardware rankings based upon the result of the process runs separately for the core and large generic accessibility control set. This will allow the user to understand the best equipment for more simple games as well as the best equipment for more complex games or versatile usage.

7. An Optimal Hands-Free Gaming Experience: My Contribution

There are some options available for hands-free gaming now, but there are plenty of improvements to be made for game and controller accessibility. My project directly helps contribute to this field by suggesting a novel way to view games, gaming technology and gaming abilities that could make video games more accessible, particularly for hands-free gaming.

7.1 Challenging Normativity

My project pushes back on one of the most prevalent commonalities of all games – the requirement of using one's hands. Hands-free gaming is by no means the most popular form of gaming, nor is it considered "normal" to use head gestures instead of classic console controllers. Further, gaming culture has become comfortable assuming that players with disabilities are responsible for their own gaming experience. This means researching and buying new technologies and controllers as well as inventing entirely new devices and workarounds. To struggle to play is the norm for a disabled gamer. My project rejects the notion that gamers are expected to invent and re-invent control systems to accommodate inaccessible game design. I challenge game designers to step up to the plate and unlock their controls systems – make their APIs easy to access and filled with all the information a gamer would need to remap all the actions in the game to new controllers.

While discussing this project with friends and family, one of the most common inquiries I received is how anyone could possibly play a game using facial gestures. It occurred to me that there is a fundamental misunderstanding surrounding hand controllers—that is, that hand controllers are inherently intuitive and easy to use. In fact, they are not! This is no more a matter of science than stating that learning to tie one's shoes is not intuitive, nor is writing, playing guitar, or using a smartphone. Everything we do with our hands is learned, usually at the cost of hours of practice. Video game hand controllers are particularly difficult to master. Even life-long gamers complain of sinking dozens of hours into learning game controllers that are only slightly different from those they use regularly.

All of this is to say that there is nothing outlandish about using one's face, feet, head, or anything else to play a video game. It is a matter of learning to use certain gestures with greater precision over time, just like hand controllers. Challenging the notion that hand controllers are the only "easy" or worth-while controllers to learn will help gamers and game designers consider other possibilities, making gaming more accessible.

Consider another art: dancing. Traditionally, dancing is a full-body activity, using muscles from the feet up to the head. Dancing usually requires a bit of space so one may dance unobstructed. Belgium dancer Annika Verplancke wished to challenge the normativity of needing space and full-body movement by performing "face dancing."[1] She posted several videos of her original face dances on her Instagram account, encouraging her followers to try as well. The videos feature her from neck-up, using only her face and head to move to the beat of the music. This is, in effect, how I hope to have challenged video game normativity in this thesis – by demonstrating that many of the boundaries of game controllers are purely imaginary.

7.2 Focusing on Universal Design

When one starts to consider improvements for accessibility, one will naturally encounter universal design principles. Even though my thesis does not specifically address each time a universal design principle is present, these guides are nonetheless the backbone of many of my arguments and my interface software concept. One of the ways my project contributes to the gaming industry is by encouraging and supporting the use of these guides through both academic and empirical research. For the purposes of this section, I will briefly visit a few of the universal design principles (as they are stated by the National Disability Authority's Centre for Excellence in Universal Design[2]) that I find the most relevant to my discussions. These are the first, second, third, and fifth principles.

The first principle is that of equitable use, or a design that is useful to people with a wide range of abilities. I will not dive into this too deeply for fear of stating the

[1] Verplancke, Annika. "A face dance." *Instagram,* 27 May 2020.

[2] "The 7 Principles." *Centre for Excellence in Universal Design, National Disability Authority*, 2020.

obvious—my project focuses hugely on including as many gamers as possible to the community, covering a vastly greater range of abilities than common game controllers. While my main goal was to help gamers achieve a better hands-free gaming experience, both my technical project and interface software concept invite gamers of all abilities to experiment with new control systems. This is because I believe hands-free gaming technology can serve communities beyond gaming. Imagine, for example, hands-free alternatives to every-day activities such as driving or using a microwave. Software that can help someone learn about hands-free options in gaming or elsewhere is useful regardless of ability, as it opens doors to future methods and technologies. These new designs would make tasks easier for some, but for others, would make some tasks possible.

The second principle is flexibility, meaning that the product design is flexible to a wide range of abilities. Because my concepts rely on a diverse set of commercially available accessibility software and hardware, a wide range of abilities are accommodated. User comfort, precision, and accuracy is also at the forefront of my experience designs as these inform the hardware and controls recommended to the users. Flexibility in game design is also one of my major discussions, as games are often designed with only traditional hand controllers in mind. This form of leniency relates to my points about casual games and happiness, discussed in Appendix A. Casual games allow players to come and go from gameplay as they please, with little or no penalties.

This makes the games more accessible because they do not require long durations of playtime to advance and have fun.

The third principle is regarding simple and intuitive use. As a user experience designer, nothing tickles me more than a simple and intuitive solution to a complex problem. However, I cannot prove that my designs hold these virtues without user research and testing. I can at least state this – this project was my quest to determine how to eliminate unnecessary complexity from the hands-free gaming experience. I wanted to find a way to add consistency to gaming accessibility, thus the invention of the tiered user comfort and frequency of use system for controls. I wanted to relieve some of the responsibility of the user to create their own hands-free experience by relocating the work on the game developer or a software such as the API and GUI discussed in Chapter 6. Whether or not these concepts are truly user-friendly can only be speculated, but I believe they present a chance for a far simpler experience than what currently exists.

The fifth principle is tolerance for error, or a design that does not punish the user for accidents. This concept is addressed early on in the thesis with the discussions of flow, fiero, and happiness. Low stakes for failure is one of the four essential ingredients to reach a state of flow, or a joyful, immersive form of productivity. Flow is essentially optimal user experience, and that stretches to gaming, education, and beyond. One of the ways I considered this principle in designs is with the eye-tracking demo game. Free online eye-trackers offer poor accuracy, and even high-end trackers only offer precision to about the size of a quarter. I took this into consideration when making my game and writing the control recommendation. For example, the game never requires the user to hover in one place for too long because I know that the eye-trackers make it difficult to hold the curser in a small area. Even though the "idle" gesture is common in eye-tracking products, I decided to build in as much error-forgiveness as possible because I knew users could be using very low-grade eye-trackers.

Ultimately, all seven of the universal design principles are relevant to my thesis, but for brevity only the most applicable were discussed.

7.3 A Case for Partial Game Design

For decades, games have offered various modes for different difficulty settings and types of play. This helps make the same game entertaining for more age groups and for a longer period of time. The classic modes of "easy," "medium," and "hard" eventually evolved into something more complex – games that intelligently predicted the game mode that created the most fun and optimized experience for the player. For the sake of this thesis, these are "partial" games, or games that favor the success and enjoyment of the player. Partial games bend to the needs and skill-level of the player rather than requiring that players meet the demands of the game. Impartial games are non-reactive to the skills and experiences of the player, and the player cannot effect the difficulty or requirements for the game. I call these games "impartial" because they do not favor the success of the player, nor can this be changed in the settings.

One game that has seen stunning success for its partial design is Epic Games' *Fortnite*. In *Fortnite*, players are placed in an arena of 100 competitors composed of other active players and some artificially intelligent robots. The object of the game is to be the last person alive in the arena. While the game is in the shooter genre, players need not engage in much shooting to win a match, instead using various tactics including building structures, setting traps, hiding, and using vehicles to survive and eliminate other players. Further, Epic Games designed an algorithm that places players in matches with other players of a similar skill level and play style. This allows players to learn at their own pace and compete in evenly-matched battles. It also allows players who have not learned more advanced tactics, such as building structures, to engage in the game with other players who have not mastered the same skills.

There are dozens of other partial games that favor the success of the player in various ways. Some games are partial by giving the player many different ways to succeed. Other games allow players to change the parameters of the game to make it more manageable or fun for them (for example, by decreasing the number of enemies or increasing the amount of time permitted to complete a level). There are even games that offer a "sandbox" mode, where certain parameters are changed to make gameplay a free-form and casual as possible (this may be giving the player unlimited money, invincibility, maximum power-ups, etc.).

Partial games are becoming more popular as concerns about accessibility rise, but the accessibility of game controls is lagging. Some games allow players to customize major controls, although few allow full customization and virtually none offer intelligent adaptation systems. The world is distinctly lacking partial game controllers.

What does a partial game controller look like? This is described in depth in Chapter 6, although remains largely nebulous. It would likely require the cooperation of game designers, hardware designers, and users to create an experience that was fully adaptive and customized for the individual. Chapter 6 describes how an interface software and API may streamline the data collection and control optimization process, and why this may be the most realistic option for the future of control accessibility. While it may seem a daunting task to recruit all the necessary cooperation, much of the hardware and API work is already done and needs only to be optimized for user access. The reality of accessible hands-free gaming is not as unreachable as it seems. I believe one of the largest contributions my thesis brings to the gaming industry is the imagination, conceptualization, and ultimately a case for partial game controls design.

8. Conclusion

Games are an ever-increasing part of cultures and every-day lives around the globe. In this way, game designers have become an incredibly powerful and influential part of society. With great power there is, of course, great responsibility, as millions of people cannot participate in video games simply because video games have not been designed for them. Yet.

The current experience for hands-free gaming is tedious – from acquiring the proper hardware or software to match a specific game, to remapping controls or finding workarounds for controls that are too difficult to access. Some gamers ultimately find that this effort is not worth it for the gaming experience achieved. However, everything we need for adaptable, flexible, highly-accessible gaming technology already exists, from the sensors to the data. These need only be combined using the collaborative efforts of game designs, hardware engineers, and the users to produce optimized experiences.

The first step to this process is to create a common language by which to discuss controls. This is a baby-step toward uniting the gamers and designers. Thus the creation of the frequency-of-use tiered system I discussed in Chapter 4 – by categorizing controls by their relevance to gameplay, gamers and designers can start making smart pairing decisions with controls. This, paired with an open-access API of the available controls for a game, would be a massive leap for gaming accessibility, and its production would cost minimal time and money on the side of the game company.

The second step would be introducing smart hardware and software into the mix – devices used to help assess the user's abilities and determine which could be the most successful at which controls or games. This is demonstrated in a rudimentary way in my technical project (Chapter 5) as I discuss the possibility of a diagnostic eye-tracking game to determine the usefulness of a particular eye-tracker to a particular user. This is just one of many ways that hardware is vital to the user's success and comfort (for example, using one camera for an eye-tracker software as opposed to four).

The third step would be the interface software concept described in Chapter 6. This is the leap forward that I dream is hovering just above the horizon – a gaming accessibility industry that has harnessed the full cooperation of game designers, hardware designers and the userbase to serve the accessibility needs of people with a massive range of abilities. This invention sounds too good to be true, but I firmly believe that it is not an if, but when this automation will arise. As I have discussed previously, as far as technological advancement, we are already here. We have been here.

With more time and technical expertise, I would take my technical project demo further and perhaps begin to outline what the interface software would look like from a technical or visual standpoint. In a more amicable world, I might have been able to perform user tests and playtesting on my demo to collect more information on how to make the experience user-friendly. If someone were to pick up this research after me, this is the first thing I would suggest – find users, talk to them, ask them to play the demo and interview them about their dreams for accessible gaming. All other research and development should move from there.

Something I realized while embarking on this project is how easy it is for people to feel detached from problems concerning accessibility. This is somewhat natural considering most people worry about their own problems and inconveniences that pertain to them specifically. If everyone constantly worried about everyone, we would all drown in anxiety. However, what goes underappreciate is how much accessibility (or the lack of it) does pertain to us, and everyone.

As people age, certain abilities will fade and others will be lost to illness and accidents. Should an individual have the privilege of living a long life, they will inevitably face a disability of some kind. Thus, a whole generation of gamers – billion of people – will age as well, and as is nature, will lose abilities. A few decades from now, there will be millions of life-long gamers facing the challenges of accessibility in gaming, and the gaming and design industries will be responsible for producing games and controllers for wider and wider ranges of abilities. This is why topics such as hands-free gaming need to be tackled as soon as possible – it is only a matter of time before the demand for such accessibility is far greater than the speed at which such technologies can be created. These technologies will be vital to keeping gamers connected to their communities as they age. It's never too early to start working on a better future, so the best time to start thinking about these realities is now.

Ultimately, this project resulted from me doing what user experience designers do – researching a subject, listening to what users have to say, and illustrating what I believe could be the best way to serve the users. The UX designer is, after all, an advocate of the people. However, this is where I must leave you – with a dream of an idea for a more accessible future in gaming. I have no such expertise to create software or hardware, and I have run out of time to delve further into the research. I hope this thesis can serve as an elevator pitch as to why hands-free gaming technology is an invention worth one's time and effort. It was certainly worth mine.

Appendix A : Casual Games' Accessibility-Friendly Design

Casual games, such as indie hit *Stardew Valley*, farming simulation *Harvest Moon*, and Nintendo's popular social simulation game *Animal Crossing: New Horizons*, have a reputation of helping players relax and unwind. Gaming researcher Jasper Juul describes a casual game as, "the design does not need to be segmented or repeatable, but the player should be able to easily interrupt the game without serious consequences and return to the session later."[1]

Generally speaking, casual games have simple gameplay and user interfaces, making them easy to understand and playable for most age groups. The Casual Gaming Association even defines casual games as "fun and easy to learn and play" on their archived website.[2] This is part of what made casual games a popular choice for those with motion disabilities -- the simplicity of the UI allows for seamless gameplay using very few controls. Casual games are also marked by their ability to play in short sessions because of their aforementioned interruptibility.

Casual game interruptibility offers players a chance to engage in a daily satisfying activity with minimal commitment, or minimal reason to sacrifice other activities. However, not all casual games are truly interruptible. A truly interruptible

[1] Jasper Juul, *A Casual Revolution Reinventing Video Games and Their Players*

[2] www.casualgamesassociation.org, accessible through the WayBackMachine

game allows the player to save their progress and quit at any point, or otherwise offers no consequence for the player suddenly quitting the game.

For example, in *Animal Crossing: New Horizons*, a player is always one press of a button away from saving and exiting the game. No opportunities will be lost if the player returns at a later time. *Animal Crossing: New Horizons* even auto-saves during gameplay, so potential progress lost from quitting the game unexpectedly is minimal. The game lacks time-sensitive quests, allowing the player to progress at their own pace.

By contrast, *Stardew Valley* has only one save point-- the player's virtual bed. If a player does not go to their bed to save before quitting, their progress for the day will be lost. Granted, one "day" *in Stardew Valley* lasts only about 15 minutes, minimizing the amount of progress that can be lost as a player is forced to save exactly one time per day (at the end of each day, when they go to bed). A player cannot save without ending the day, so there is a non-zero opportunity cost for interrupting the game mid- day to save. The consequence for not saving the game or for saving mid-day (therefore ending the day) is fairly minimal, but still more present than *Animal Crossing: New Horizons*, making *Stardew Valley* less interruptible.

Casual games and their interruptibility are highly compatible with a happy life, according to McGonigal's four secrets to happiness. This is mainly due to the third secret: social connection.

Consider the following excerpt from McGonigal's *Reality is Broken:*

> "More gaming by more people is the primary goal of the industry. But the industry wants to create lifelong gamers: people who can balance their favorite games with full and active lives... Beyond a certain playing threshold—for most gamers, it seems to be somewhere around twenty hours a week—[gamers] start to wonder if they're perhaps missing out on real life." [3]

People who play video games too often or who become addicted to games will certainly start to miss out on real life. While many games allow gamers to connect with other people online, this is not what McGonigal is referring to in her third secret -- she clarifies that we gain "a large percentage of our happiness from spending time with the people we care about."[4]

Interruptibility is not the magic-bullet solution to video game addiction, but casual gaming certainly has a case for keeping people involved in real life. Casual games are not designed to be addicting. They are designed for short bursts of low-stakes tasks and entertainment. They are too simple to be fun to play all day. The utterly low commitment to continue gameplay makes putting down the game to join your family at the dinner table much easier. It makes the game something that can be as ornamental to your life as reading a morning newspaper or watching a TV show before bed.

Part of casual game interruptibility means that the game isn't gripping enough to cause players to miss out on real life.

[3] Jane McGonigal, *Reality is Broken*, p. 43.
[4] Jane McGonigal, *Reality is Broken*, p. 49.

Appendix B : Super Smash Bros. Playable Character Gender Analysis

While studying at Duke, I conducted a brief research project on the *Super Smash Bros.* series to explore how Nintendo has diversified (or failed to diversify) the playable characters' genders in the *Super Smash Bros.* series over time.

The *Super Smash Bros.* (SSB) video game series was created by Masahiro Sakurai and published by Nintendo. While SSB is known to have many playable characters, none of these characters are unique to only SSB games. SSB is actually a conglomerate of major characters from many other popular games. The object of SSB is to fight characters (either played by other people or controlled by the computer) to complete challenges and work through stories. SSB has five games in the series and is playable on seven different consoles. Today, tens of millions of copies of SSB have been sold worldwide and there is a large competitive community dedicated to the game.

Before I get started on my analysis, there are two things I would like to clarify. The first is that the gender of characters is not always clear. I will go over how I determined the genders of the characters shortly, but it is important to realize that these characters are fictional and not always intended to take on qualities such as gender, race or sexuality. The second is that I adore Nintendo and the *Super Smash Bros.* series. It is with the utmost respect and scholarly curiosity that I choose this game for my analysis.

In an effort to make my analysis as simple yet accurate as possible, I have broken down the characters into four major gender categories:

Male: For characters that have been identified by their franchise as male.

Female: For characters that have been identified by their franchise as female.

Player's Choice: For characters that could be male or female depending on the choice of the player.

Ambiguous: For genderless characters, or characters who have genders but the gender is not specified. For example, Pokémon are often gendered in the Pokémon series, but whether the player is using a female or a male Pikachu (for example), is not specified in SSB. Ambiguous was also assigned to teams of characters where there was a mix of male and female characters. For example, there is one female and one male Ice Climber. The Ice Climbers were marked as ambiguous because these characters are used together, as a "one character" unit.

Before I continue, here is a review of some important terminology:

Playable character: These are characters that the player can control during regular gameplay. This is opposed to NPCs, or non-playable characters who cannot be controlled by the player.

DLC: Stands for Downloadable Content. This is extra content that was not included in the purchase of the original game. In the SSB series, players have to pay for DLC, but it is not a reoccurring subscription. Characters are added to the SSB games through DLC. The DLC characters available as of October 13, 2019 are counted in my gender analysis.

That's enough preamble. On to the good stuff.

The first chart I created shows the number of characters by gender in each SSB game. The names of the games have been abbreviated in the chart image, but "SSB" stands for the original *Super Smash Bros.* for the Nintendo 64. Note that you can see this chart and all of the following charts in their interactive form by visiting my personal website, linked in the footnotes of this page.[5]

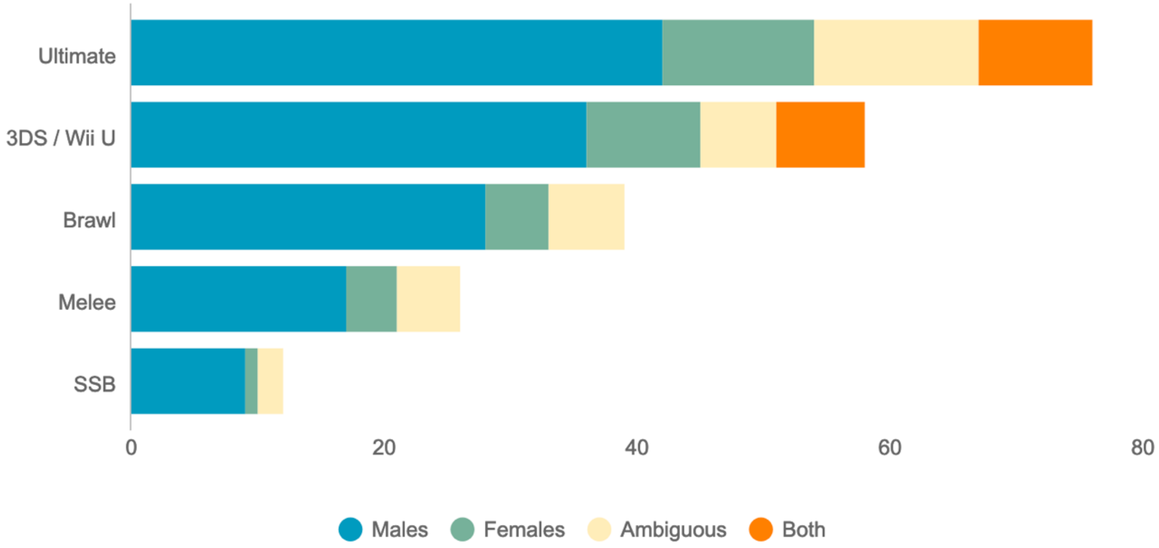


Figure 38: SSB Playable Character Gender Breakdown

[5] https://www.andreajoybrucculeri.com/supersmashanalysis

With only the chart displayed above, it is difficult to determine whether or not Nintendo successfully diversified the genders of their playable characters over time. Because of this, I made a second interactive chart to show the percentage of each gender category.

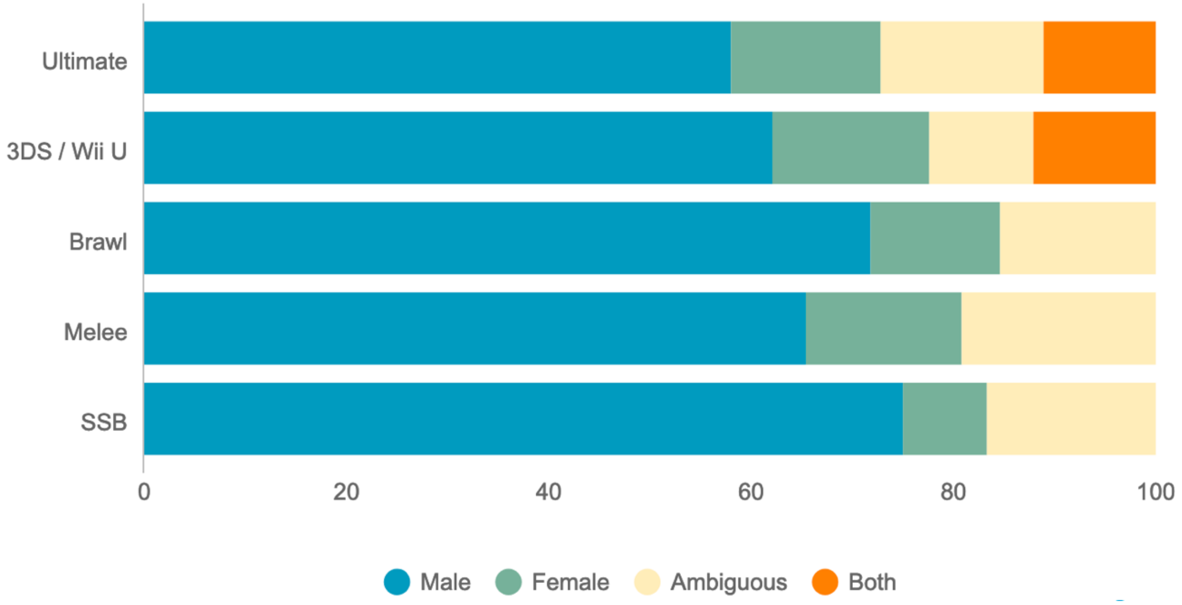


Figure 39: SSB Playable Character % Gender Breakdown

Nintendo has steadily improved the gender ratio of the SSB playable characters, and this chart may not give justice to how dramatic of an improvement has been made.

The original SSB game was 75% male, and the most recent SSB game is roughly 58% male. That's a 22.6% decrease, which is impressive considering Nintendo mainly has male-dominated franchises to pick from. Nintendo can't necessarily solve the problem by removing male characters– only by adding female character options and diluting the pool. I also support Nintendo's decision to add more "both" characters to the game. Instead of adding only females, Nintendo is giving everyone more options to find characters they identify with by having multiple genders (even non-binary options) available as alternative forms of the same character.

When I originally completed this project, I stopped at this point and determined that Nintendo was making an honest effort to include more non-male characters in *Super Smash Bros.* However, I decided that this painted an incomplete picture of the progress made on this subject. There is more than what meets the eye.

Although there are technically more non-male playable characters included in SSB, I couldn't help but feel that the character roster still looked overwhelmingly masculine. This prompted the next step in this project: determining if Nintendo had successfully included more playable characters that appeared non-male.

Admittedly, I used a less scientific process than when I was figuring out how many characters were definitively male, female, or neither. Whether or not a character looks like a male is largely an opinion and often relies on stereotypes. However it is through the lens of these stereotypes and preconceived notions that people make determinations about others such as their age, gender, socioeconomic status, and so on. For brevity I will not explain why this is true-- those who wish to read more on this subject I will refer to Malcolm Gladwell in his book Blink.[6]

[6] Gladwell, Malcolm. *Blink: the Power of Thinking without Thinking*.

I judged whether or not a character looked like a male, female or ambiguous based on a few visual features:

Color: Characters who were or had clothes in more pastel colors, pinks and purples were considered more feminine. Blues, reds and greens were considered to be more masculine colors, and some colors such as yellow, white and black were considered neutral.

Clothing: Characters wearing dresses, skirts, lace or frills were considered to look feminine. Characters who wore pants but no shirt, or heavily-clad armor were considered to look masculine. Most other clothing was considered neutral.

Bodily features: Characters were considered female if they appeared to have female body parts (such as breasts and wide hips), facial features typically associated with female faces, or long hair. Characters were considered male if they appeared to have a body structure associated with males (such as extremely bulky muscles or a V-shaped figure), facial features typically associated with male faces, or short hair.

Name: Every character in SSB has a name listed under their picture. Names traditionally assigned to men in Western culture were considered to be masculine, and vise versa for women.

I marked each character as appearing male, female, or ambiguous on the same spreadsheet mentioned previously. I did my best to mark as few characters as possible as ambiguous based on my criteria. This was an imperfect process and there were some characters I struggled to label. However I made my decisions in the most unbiased and scientific way I could by following the above criteria. The following charts display the results of my efforts.

The first chart displays the raw numbers of playable characters as the gender they appear to be in each series.

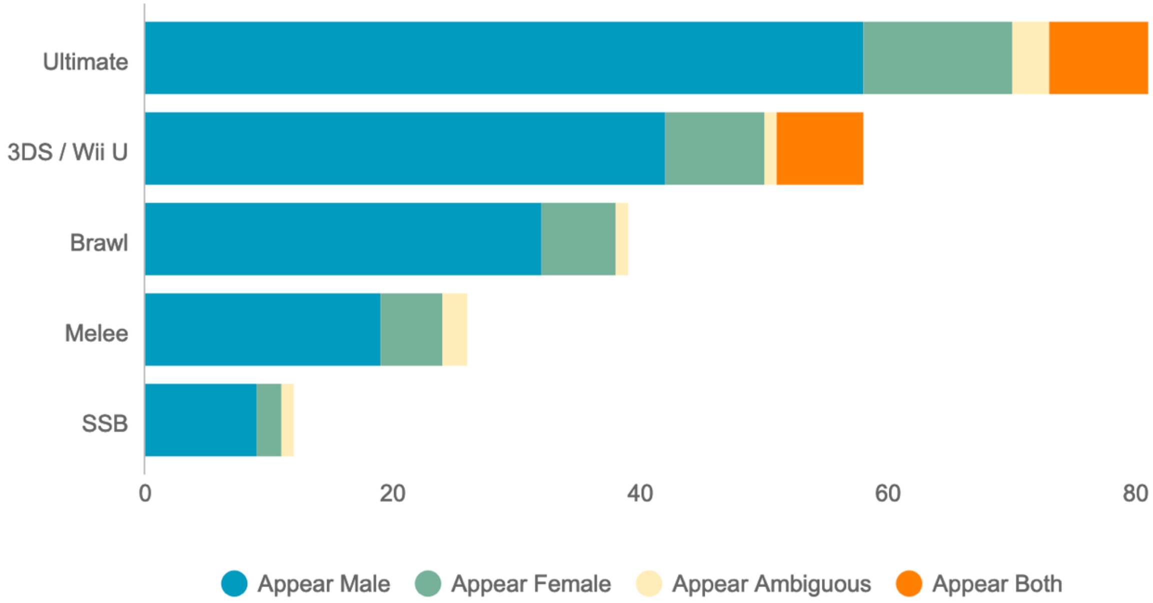


Figure 40: SSB Playable Character Gender Appearance Breakdown

Again, it is difficult to tell from this graph if there was an improvement in Nintendo adding female-appearing characters over time. I created a second graph to show the same data by percentage to make the results more clear.

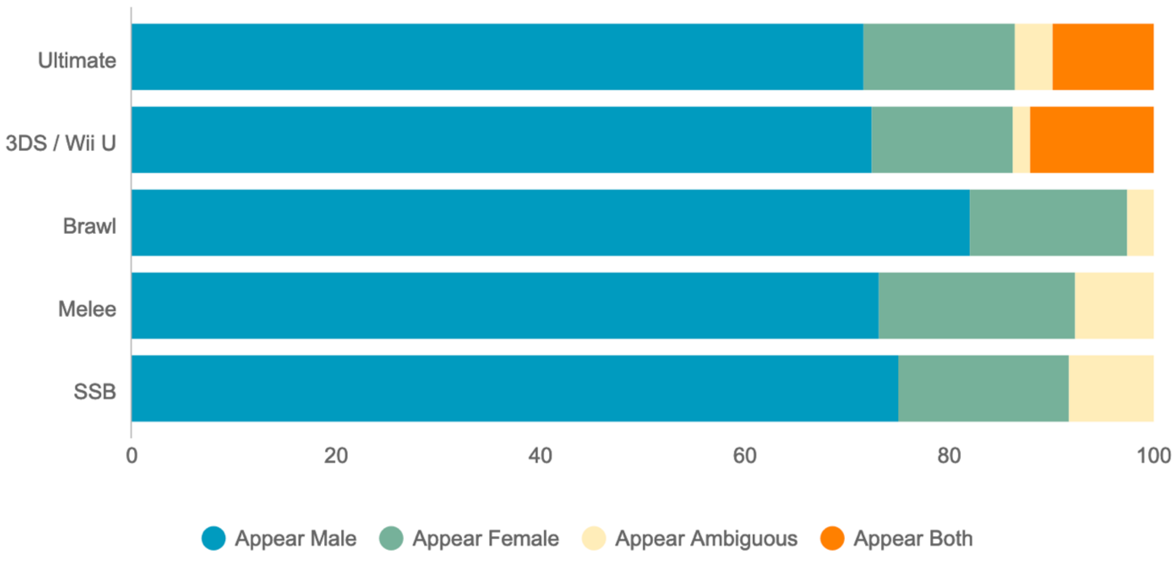


Figure 41: SSB Playable Character % Gender Appearance Breakdown

While there are more characters that are identified by their franchise as female, the *Super Smash Bros.* playable character roster is still an incredibly masculine-looking space. There has been only a very small improvement over time, with the original SSB appearing 75% male and the most recent edition of the game appearing 71.6% male. Whether or not this difference is significant enough to be noticeable to players is debatable.

With such a noticeable difference in Nintendo advertising over time, I was hoping to see a more significant change in the characters. I will note that pairing all four graphs together does suggest that Nintendo is being more inclusive to non-male characters by making somewhat masculine or androgynous female characters. For the scope of this project, I will not discuss whether or not this is a bad thing for the female gaming community. However I will end on this question -- is making a more gender-inclusive video game environment a struggle to invite women into gaming more actively, or a struggle to make men accept that women are already there?

Bibliography

AbleGamers Charity. "AbleGamers and Gamers Outreach Partner to Help Children with Disabilities in Need." *The AbleGamers Charity*, 2017, ablegamers.org/ablegamers-and-gamers-outreach-partner-to-help-children-with-disabilities-in-need/.

Avedon, Elliott M., and Brian Sutton-Smith. *The Study of Games*. Wiley, 1971.

Bailenson, Jeremy. *Experience on Demand: What Virtual Reality Is, How It Works, and What It Can Do.* W. W. Norton &amp; Company, Inc., 2019.

"Casual Games Association." CGA :: Casual Games Association, Feb. 2009, web.archive.org/web/ 20090217221337/casualgamesassociation.org/faq.php#casualgames.

Cote, Amanda C. "Writing 'Gamers.'" *Games and Culture*, vol. 13, no. 5, 2015, pp. 479–503., doi:10.1177/1555412015624742.

Csikszentmihalyi, Mihaly. *Flow: the Psychology of Optimal Experience.* Harper Row, 2009.

Davis, Lennard J. *The Disability Studies Reader*. Routledge, 2017.

Dotsenko, Andrew. "Designing Game Controls." *Game Design Framework*, 28 Mar. 2017, gamedesignframework.net/designing-game-controls/.

Downs, Edward, and Stacy L. Smith. "Keeping Abreast of Hypersexuality: A Video Game Character Content Analysis." *Sex Roles*, vol. 62, no. 11-12, 2009, pp. 721–733., doi:10.1007/s11199-009-9637-1.

Editorial Team. "Global Video Game Consumer Population Passes 3 Billion." *DFC Intelligence*, DFC Intelligence, 14 Aug. 2020, www.dfcint.com/dossier/global-video-game-consumer-population/.

Gee, James Paul. *Unified Discourse Analysis: Language, Reality, Virtual Worlds and Video Games.* 1st ed., Routledge, 2014.

Gee, James Paul. *Why Video Games Are Good for Your Soul: Pleasure and Learning*. Common Ground Publishing, 2005.

Gladwell, Malcolm. *Blink: the Power of Thinking without Thinking*. Back Bay Books, 2019.

Glaubke, Christina R, et al. "Fair Play? Violence, Gender and Race in Video Games." *Children Now*, Oakland, CA., 2001.

Holmes, Kat. *Mismatch: How Inclusion Shapes Design*. The MIT Press, 2020.

"International Game Developers Association." *IGDA*, 2020, igda.org/.

Jerald, Jason. *The VR Book: Human-Centered Design for Virtual Reality*. ACM Books, 2016.

Juul, Jesper. *A Casual Revolution: Reinventing Video Games and Their Players*. MIT Press, 2012.

Leonard, David. "'Live in Your World, Play in Ours': Race, Video Games, and Consuming the Other.*" SIMILE Studies In Media &amp*; Information Literacy Education , vol. 3, no. 4, Nov. 2003, doi:10.3138/sim.3.4.002.

Leonard, David. "Performing Blackness: Virtual Sports and Becoming the Other in an Era of White Supremacy." Re:Skin, by Mary Flanagan and Austin Booth, MIT Press, 2006, pp. 321–322.

Lien, Tracey. "No Girls Allowed." *Polygon*, Vox Media, 2 Dec. 2013, www.polygon.com/features/2013/12/2/5143856/no-girls-allowed.

Malkowski, Jennifer, and TreaAndrea M. Russworm. *Gaming Representation Race, Gender, and Sexuality in Video Games.* Indiana University Press, 2017.

McGonigal, Jane. *Reality Is Broken: Why Games Make Us Better and How They Can Change the World*. Vintage, 2012.

Miller, Hawken. "'It's My Escape.' How Video Games Help People Cope with Disabilities." *The Washington Post*, WP Company, 10 Dec. 2019, www.washingtonpost.com/video-games/2019/10/14/its-my-escape-how-video-games-help-people-cope-with-disabilities/.

Pierre-Louis, Stanley. "ESA Leadership Desk: 214 Million Americans Play Video Games." *Entertainment Software Association*, Entertainment Software Association, 15 July 2020, www.theesa.com/perspectives/214-million-americans-play-video-games/.

Russworm, TreaAndrea M. "Computational Blackness: The Procedural Logics of Race, Game, and Cinema, or How Spike Lee's Livin' Da Dream Productively 'Broke' a Popular Video Game ." *Black Camera*, vol. 10, no. 1, 2018, pp. 193–212.

Salen, Katie, and Eric Zimmerman. *Rules of Play: Game Design Fundamentals*. The MIT Press, 2010.

Shen, Szu-Ching. "Incidence, Risk, and Associated Factors of Depression in Adults with Physical and Sensory Disabilities: A Nationwide Population-Based Study." *US National Library of Medicine*, 2017, doi:10.1371.

Stone, Jeffrey A. "Self-Identification as a 'Gamer' among College Students: Influencing Factors and Perceived Characteristics." *New Media & Society*, vol. 21, no. 11-12, 2019, pp. 2607–2627., doi:10.1177/1461444819854733.

Stoner, Grant. "How Accessibility Consultants Are Building a More Inclusive Video Game Industry behind the Scenes." *The Washington Post*, WP Company, 25 Feb. 2020, www.washingtonpost.com/video-games/2020/02/25/how-accessibility-consultants-are-building-more-inclusive-video-game-industry-behind-scenes/.

"The 7 Principles." *Centre for Excellence in Universal Design, National Disability Authority,* 2020, universaldesign.ie/What-is-Universal-Design/The-7-Principles/.

Turkle, Sherry. *The Second Self: Computers and the Human Spirit.* MIT Press, 2005. Goodley, Dan. Disabilities Studies. 1st ed., SAGE Publications Ltd, 2010.

United States*, Department of Labor, Bureau of Labor Statistics, and Press Office*. Persons with a Disability: Labor Force Characteristics -- 2019, Bureau of Labor Statistics, 2020, pp. 1–2.

Verplancke, Annika. "A face dance." *Instagram,* 27 May 2020, https://www.instagram.com/p/CAsfZolAHRj/.

Wardrip-Fruin, Noah. "Expressive Processing: Digital Fictions, Computer Games, and Software Studies." *Literary and Linguistic Computing*, vol. 26, no. 4, 2011, pp. 299-352., doi:10.1093/llc/fqr035.

Wardrip-Fruin, Noah, et al. *First Person: New Media as Story, Performance, and Game.* The MIT Press, 2006.

Wynne, Kelly. "Here's Why 'Animal Crossing' Is Great for Your Mental Health Right Now." *Newsweek*, 26 Mar. 2020, www.newsweek.com/animal-crossing-new-horizons-great-your-mental-health-through-covid-19- professionals-say-1494525.