

An Exploration of In-Game Action Mappings with a Deformable Game Controller

by

Paden Shorey

A thesis submitted to the Faculty of Graduate and Postdoctoral Affairs in
partial fulfillment of the requirements for the degree of

Masters of Applied Sciences

in

Human Computer Interaction

Carleton University
Ottawa, Ontario

© 2016, Paden Shorey

Abstract

Deformation input consists of methods of interaction that make use of material deformation in order to control on-screen variables. In this thesis, we explore controller input mappings for games using a deformable prototype that combines deformation gestures with standard button input. In experiment one, we tested discrete gestures using three games. We categorized the control schemes as binary (traditional), action, and navigation, the latter two named based on the game mechanics the gestures were mapped to. We found that the binary scheme performed the best, but gesture-based control schemes are stimulating and appealing. Results also suggest that the deformation gestures are best mapped to simple and natural tasks. In experiment two we tested continuous gestures in a 3D racing game using the same control scheme categorization. Results were mostly consistent with experiment one but showed an improvement in performance and preference for the action control scheme.

Acknowledgments

There are so many people I would like to acknowledge. First of all, my parents. Without their continual nagging I would have completely forgotten that I was in school writing a thesis. Their constant questions regarding my progress frustrated me just enough to actually convince me to get some work done. But seriously, thank you both Mom and Dad for being there always and listening to me vent about my consistently high stress levels, I could not have done this without your support.

I would like to thank my girlfriend Christine for encouraging me through these past two years and for being someone to complain to about all my trials and tribulations. You really seem to understand my problems that can barely be considered problems in the first place. Thank you for not taking anyone's side but mine when things seemed impossible and for giving me the breaks I needed on the weekends. You made this process a lot easier on me.

I would like to thank my office partners, Eli, Nick, and Alex, for dealing with me over the past two years. Thank you for letting me share an office with you, bringing me to the gym to relieve some stress, and sharing stories completely unrelated to anything we're supposed to be doing at school. I definitely could have had worse office mates. I would like to thank all my friends (you know who you are ... but you'll never read this) for caring about what I'm doing in school and for dealing with my daily complaints. I also want to thank them for all of their suggestions regarding my thesis even if I didn't use any of them.

Finally, I would like to thank my amazing supervisor, Dr. Audrey Girouard, for guiding me through this process and giving me this priceless opportunity. I definitely could not have done this without her ... seriously. Her patience, understanding, and expertise made this process exponentially easier than it could have been. She's the best thesis supervisor imaginable.

Table of Contents

| | |
|--|-------------|
| Abstract..... | ii |
| Acknowledgments | iii |
| Table of Contents | iv |
| List of Tables | vii |
| List of Figures..... | viii |
| Chapter 1. Introduction..... | 1 |
| 1.1. Overview..... | 1 |
| 1.2. Contributions | 3 |
| 1.3. Outline of Thesis | 4 |
| Chapter 2. Related Work | 5 |
| 2.1. Deformation Interactions..... | 5 |
| 2.2. Deformation Interactions With Games | 8 |
| 2.3. Deformation Interactions With Other Input | 11 |
| 2.4. Novel and Natural Game Interaction | 13 |
| Chapter 3. Prototype..... | 17 |
| 3.1. Interaction Language | 17 |
| 3.2. Hardware..... | 19 |
| 3.3. Software..... | 23 |
| Chapter 4. Experiment 1: Flexible Input Methods for Classic Video Games | 27 |
| 4.1. Overview..... | 27 |
| 4.2. Games | 28 |

| | | |
|---|---|------------|
| 4.3. | Control Schemes | 31 |
| 4.4. | Methods | 33 |
| 4.5. | Hypotheses..... | 35 |
| 4.6. | Participants | 36 |
| 4.7. | Results & Analysis..... | 36 |
| 4.8. | Discussion | 55 |
| 4.9. | Summary | 62 |
| Chapter 5. Experiment 2: Continuous Flexible Input Methods in 3D Racing Games | | 64 |
| 5.1. | Overview..... | 64 |
| 5.2. | Game..... | 65 |
| 5.3. | Control Schemes | 68 |
| 5.4. | Hypotheses..... | 70 |
| 5.5. | Methods | 71 |
| 5.6. | Participants | 72 |
| 5.7. | Results & Analysis..... | 73 |
| 5.8. | Discussion | 85 |
| 5.9. | Summary | 91 |
| Chapter 6. General Discussion..... | | 93 |
| 6.1. | Comparing the Results of our two Experiments..... | 93 |
| 6.2. | Limitations | 96 |
| Chapter 7. Conclusion | | 99 |
| 7.1. | Overview..... | 99 |
| 7.2. | Future Work | 101 |
| Chapter 8. Bibliography | | 103 |

| | |
|----------------------------------|------------|
| Chapter 9. Appendix..... | 111 |
| 9.1. Consent Form..... | 111 |
| 9.2. Questionnaires | 115 |

List of Tables

| | |
|--|----|
| Table 1. Donkey Kong Control Schemes | 32 |
| Table 2. Mike Tyson’s Punch-Out Control Schemes | 32 |
| Table 3. Tetris Control Schemes..... | 33 |
| Table 4. Overview of Performance Statistics for Experiment 1 | 37 |
| Table 5. Summary of Game Ranking Statistics for Experiment 1..... | 41 |
| Table 6. Summary of UEQ Statistics for Experiment 1..... | 46 |
| Table 7. Total Input Counts Across All Participants | 53 |
| Table 8. Summary of Performance Statistics for Experiment 2 | 73 |
| Table 9. Summary of Trial Experience Statistics for Experiment 2 | 78 |
| Table 10. Summary of UEQ Statistics for Experiment 2..... | 82 |

List of Figures

| | |
|---|----|
| Figure 1. Namco’s neGcon Controller (Namco Bandai Games, 2015) | 16 |
| Figure 2. Nintendo Entertainment System Controller | 17 |
| Figure 3. The Four Possible Controller Gestures; (1) Bend Up (2) Bend Down (3) Twist Left (4) Twist Right..... | 18 |
| Figure 4. Controller Dimensions Top-Down | 19 |
| Figure 5. Controller Dimensions Front View | 20 |
| Figure 6. Front side of Prototype for Study 1 | 21 |
| Figure 7. Backside of Prototype for Study 1..... | 21 |
| Figure 8. Front side of Prototype for Study 2 | 22 |
| Figure 9. Backside of Prototype for Study 2..... | 22 |
| Figure 10. Apparatus, Experiment 1 | 25 |
| Figure 11. Apparatus, Experiment 2 | 26 |
| Figure 12. Donkey Kong (Miyamoto, 1981) | 29 |
| Figure 13. Mike Tyson’s Punch-Out (Genyo, 1987) | 30 |
| Figure 14. Tetris (Pajitnov & Pokhilko, 1984) | 31 |
| Figure 15. Control Schemes for Experiment 1 | 33 |
| Figure 16. Performance Mean Scores with Standard Deviation and Significant Differences (black horizontal lines) | 38 |
| Figure 17. Performance Mean Max Level with Standard Deviation and Significant Differences | 39 |
| Figure 18. Performance Mean Death with Standard Deviation and Significant Differences..... | 40 |
| Figure 19. Scheme Rankings, Most Fun..... | 42 |
| Figure 20. Scheme Rankings, Most Comfortable with Significant Differences..... | 43 |

| | |
|---|----|
| Figure 21. Scheme Rankings, Most Frustrating with Significant Differences | 44 |
| Figure 22. Scheme Rankings, Most Natural | 45 |
| Figure 23. UEQ Mean Scores for Donkey Kong with Standard Deviation and Significant Differences | 47 |
| Figure 24. UEQ Mean Scores for Mike Tyson’s Punch-Out with Standard Deviation and Significant Differences | 48 |
| Figure 25. UEQ Mean Scores for Tetris with Standard Deviation and Significant Differences .. | 50 |
| Figure 26. NASA TLX Total Workload and Significant Differences | 51 |
| Figure 27. Vote Totals for Most Fun Game..... | 52 |
| Figure 28. Total Input Count | 53 |
| Figure 29. Twist Input Ratios | 54 |
| Figure 30. In-Game View for the Speed Track..... | 66 |
| Figure 31. In-Game View for the Precision Track | 67 |
| Figure 32. Aerial View of the Speed (Top) and Precision (Bottom) Tracks | 67 |
| Figure 33. Control Schemes for Experiment 2 | 69 |
| Figure 34. Mean Time for Speed Track with Standard Deviation..... | 74 |
| Figure 35. Mean Boost Usage for Speed Track with Standard Deviation..... | 75 |
| Figure 36. Mean Collision Count for Speed Track with Standard Deviation..... | 76 |
| Figure 37. Mean Time for Precision Track with Standard Deviation..... | 77 |
| Figure 38. Mean Collision Count for Precision Track with Standard Deviation..... | 77 |
| Figure 39. Track-Independent Perceived-Performance Rankings | 79 |
| Figure 40. Track-Independent Naturalness Rankings..... | 79 |
| Figure 41. Track Independent Fun Rankings..... | 80 |

| | |
|---|----|
| Figure 42. Track-Independent Frustration Rankings | 81 |
| Figure 43. Overall UEQ Scores | 82 |
| Figure 44. Scheme Preference Rankings Speed Track | 83 |
| Figure 45. Scheme Preference Rankings Precision Track | 83 |
| Figure 46. Track Preference by Scheme | 84 |
| Figure 47. Total Votes for Most Fun Combination | 85 |
| Figure 48. Total Votes for Least Fun Combination | 85 |

Chapter 1. Introduction

1.1. Overview

We have been playing video games with the same standard controllers for decades. Developers build most games around the same interaction methods: directional pads, analogue sticks, buttons, and triggers. Alternatively, game companies and researchers have been introducing novel methods of input into the global gaming market. A few examples of the more successful novel controllers include the Dance Dance Revolution mat, the Xbox Racing Wheel, Tony Hawk Ride's skateboard, and Harmonix's RockBand instrument peripherals. Developers create most of these novel controllers with a specific use in mind and they are not often meant for use with a broad range of video games. Some commonplace novel controllers that exist include Nintendo's Wiimote, Microsoft's Kinect, Leap Motion, and the Wii Balance Board. These controllers were designed with a more general sense of use in mind. Researchers have had success in the past with various natural user interfaces. We suggest that combining traditional input, such as the buttons found on a standard controller, with a method of input capable of more natural interactions (i.e. bend and twist gestures), has the potential to demonstrate some interesting results.

New methods of input are constantly being researched and developed, especially in regards to gaming. One area that has been introduced but not looked at in depth is deformation gestures in games: only a few researchers have used bending and twisting as methods of input for video games: Herkenrath et al.'s TWEND (2008), Ye and Khalid's Cobra (2010), and Lo's Bendy (2013), which all focus on bending as game input. However, these projects focused solely on deformation and did not make use of any other forms of input. With Cobra and Bendy, the games were played on the devices themselves using either flexible displays or projection onto the

flexible device. As many game controllers use a standard, external display, we hypothesize that users might appreciate evaluating the use of deformation gestures on a stand-alone controller. We also believe that when testing a stand-alone controller, we can avoid common issues found in mobile game such as the fat-finger problem (Siek, Rogers, & Connelly, 2005). We further propose that combining bending and twisting with the standard forms of input, such as buttons and directional pads, could yield interesting results. We believe that the implementation of deformation gestures into pre-existing games could increase performance when mapped properly to in-game actions. We also believe that it would make the experience more stimulating and refreshing to players who are used to playing games with standard controllers. Research suggests that a large number of deformation gestures exist with a device as simple as a flat piece of flexible material with bend sensors (Lee et al., 2010). Introducing upwards of 32 new gestures to users can be overwhelming and cause them to get frustrated and perform worse. We believe that providing them with a combination of fewer new gestures as well as methods of input most users are familiar with, such as buttons, could yield better and more immediate results.

Bend and twist input, combined with standard methods of input such as buttons, have the potential to make video games more fun and easier to play when mapped correctly to in-game mechanics. This hypothesis is based on the continual use of standard methods of input within the gaming industry today combined with the novel and natural types of controllers that researchers have been developing and receiving positive feedback towards. Implementing gestures parallel to button input provides users with more input options that are easily accessible without needing to move fingers away from buttons. Limiting ourselves to bend and twist provides us with four unique deformation gestures. As there is almost an endless number of mapping combinations that can be made between a video game controller and a game, we need to look into some common

factors within games and make educated guesses on which types of actions will map best to buttons and gestures for any given game. To explore these possible mappings, we test three unique, but simple, arcade games and develop three control schemes based on two generic in-game mechanics: action and navigation. Our results led us to more complex games that could make better use of gestures in a more natural way. Our second experiment again focuses on combined control mappings but with a 3D racing game using continuous gestures as opposed to discrete gestures as used in experiment one. We conclude by suggesting ways of mapping gestures to in-game mechanics and stating ways in which a similar prototype could be improved.

1.2. Contributions

Our main contributions are as follows:

1. Implementing a standalone controller that makes use of deformation gestures.
2. Combining deformation input with standard button input.

Our inclusion of buttons similar to those found on a classic game pad is a new concept that has yet to be studied. We create a standalone device with an external display instead of implementing the display on the device itself as most prior work has done (Kildal, Paasovaara, & Aaltonen, 2012; Lo, 2013; Ye & Khalid, 2010). The prototype sensor placement, and how they were secured to the device, introduce an easy way to detect both bends and twists with an acceptable degree of reliability using only two sensors. Our exploration of numerous game mechanics and in-game actions, using discrete and continuous gestures, gives a general idea of which types of actions are best mapped to bends and twists. We list several design recommendations based on our results.

1.3. Outline of Thesis

This thesis is divided into seven chapters. Chapter one introduces the topic of the thesis and reveals how we plan on exploring it. In chapter two, we take a look at some work done by other researchers regarding deformation interactions and how they mix with gaming. We will also look at some work concerning deformation input combined with other inputs, and novel controllers.

Chapter 3 describes our prototype from a hardware and software perspective and discuss some of the issues and modifications that needed to be made throughout the design process and experimentation.

In chapter 4, we discuss our first experiment, flexible input methods for classic video games, and describe how we ran it and why we chose to take this route. We look at the results and discuss our finding.

Chapter 5 looks at experiment two, continuous flexible input methods in 3D racing games, and justify our design and methods. We again take a look at the results and discuss our findings.

In chapter 6 we take an overall look at the similarities between experiment one and experiment two. We make some design recommendations based on our results and we cover some limitations faced throughout the design and experimentation process.

Finally, in chapter 7, we conclude with some discussion regarding our thesis statement and present some ideas for future work.

Chapter 2. Related Work

To design and test our controller, which uses both deformation in addition to traditional button inputs in a gaming environment, we leveraged previous works exploring deformation interactions as a sole method of input, deformation interaction in games, and deformation interaction combined with other forms of input and take lessons from other innovative video game controllers.

2.1. Deformation Interactions

The constant improvement and development of reliable methods of measuring deformation in user interfaces has opened up opportunities to researchers to nearly endless unexplored opportunities. Deformation is a broad category of interaction that includes bends, twists, wave-forms, and scrunches in the device. Researchers are testing many unique applications and are beginning to develop standards for deformation interaction including where and how to bend, and how to classify various deformation gestures.

Deformable interfaces share a subset of common interaction techniques. Lahey et al. (2011) used their prototype, PaperPhone, to test user-defined gestures and pairings using an actual flexible display. Participants held the phone in the left hand and defined a total of eight gestures to be used on the right side of the phone. They used these eight gestures to define gesture pairs. The three pairs that scored the highest were: (1) side of display, up/down, (2) top corner, up/down, and (3) bottom corner, up/down. Warren et al. (2013) classify these gestures using seven properties: location, direction, size of the bent area, angle, edge, speed of bend, and duration of bend. Their prototype resembles an 8" x10" piece of paper with a grip strip down the left side. Using this prototype, they determined that users preferred the top corner bend location, agreeing with one of the findings in PaperPhone, and that upwards bends (towards the user)

tended to be larger. They also discovered that users can distinguish up to two levels of bend magnitude. Lee et al. (2010) classified 32 gestures across three different mediums: plastic, paper, and elastic cloth. They tested these gestures by asking participants to define their own gestures for specific tasks. They noticed that their participants tended to derive their gesture ideas off of daily activities such as folding clothes. Kildal (2012) created a prototype that was designed to be held with two hands. He specified four different interaction techniques: bend up, bend down, twist up, and twist down. They classified bending up as the centre of the device being bent towards the user whereas bending down was the opposite. The classified twist down as the right side of the device being rotated towards the user and the left side of the device being rotated away from the user. Twist up was the opposite.

Schewesig et al. presented us with Gummi (2004), another deformable handheld device. Users can only deform Gummi two ways: bend up, and bend down. They also included “transition up” and “transition down” as input methods which represent the continual bend from the neutral position to the threshold where it is considered fully bent. On top of these four inputs, they implemented a “double up” and “double down” when users bend the device up or down twice in quick succession, mimicking a double-click feature on a standard mouse. They tested a variety interaction tasks including text-input, map navigation, and 2D scrolling. Although no formal experiments were performed using Gummi, observational tests demonstrated that users could learn how to interact with Gummi with only 2–3 minutes of explanation and experimentation. They state that deformation interaction with Gummi is best suited for simple tasks such as zooming in or navigating pages of text. They do not believe that bendable devices will replace more traditional methods of input. They conclude by saying that bendable interaction is feasible, effective, and enjoyable demonstrating definite potential for future work.

Herkenrath et al. (2008) created TWEND, a flexible prototype with both bend and twist input capabilities. Their prototype used eight optical bend sensors and a piece of foam layered between two pieces of flexible plastic. It did not have a screen. They managed to implement 18 unique gestures including bends and waveforms, but did not manage to implement twist due to the topological deformation of the materials. This is something we are hoping to make functional with our prototype. They also made the assumption that bending will not replace standard computer input methods such as mouse and keyboard. While they did not conduct any experiments with TWEND, they did support the idea that gesture input should not replace current input and suggested a number of unique ways of interacting with a flexible prototype.

Kildal, Paasovaara, and Aaltonen (2012) also attempted to test twist input with the Nokia Kinetic Device. They focused on a similar subset of deformation gestures as TWEND, testing only four gestures: bend down, bend up, twist up, and twist down. They looked into force and deformation range as it applies to deformable user interfaces (DUI), and factors affecting the accuracy and controllability of using these gestures in regards to user experience. They, like Herkenrath et al., knew that deformable gestures were not going to replace other methods of input such as touch or buttons, so they focused on determining what bend and twist gestures are really good for. Using data from their prior work (Kildal, 2012), they determined four things: (1) bend and twist are performed better with two hands, (2) up and down when referring to twist is intuitively different depending on the user's handedness, (3) continuous gestures are better for tasks handling the magnitude of a parameter, and (4) discrete gestures are better used to trigger discrete actions.

Researchers have used various deformation gestures for numerous applications in attempt to discover where exactly gestures fit best. Deformation interaction using paper-like deformable

interfaces is a common topic of research within this field due to the natural resemblance of a piece of paper to a thin, flexible display (Tajika, Yonezawa, & Mitsunaga, 2008; Watanabe, Mochizuki, & Horry, 2008; Wightman, Ginn, & Vertegaal, 2011). Gallant et al. (2008) developed a paper-like prototype they classified as a foldable input device (FID). They suggested many unique uses for this device such as scooping and shaking to perform tasks such as 3D modelling and document navigation. Researchers have also used deformation interaction to perform music (Troiano, Pedersen, & Hornbæk, 2015), manipulate word processing carets (Eady & Girouard, 2015), create secure passwords (Maqsood, Chiasson, & Girouard, 2013), and control a TV as a remote (Lee et al., 2011). These are all novel examples of how deformation gestures have been used in the past. Knowing where deformation gestures have been successful and where they have failed allows us to make educated choices when designing control mapping for video games.

It is important for us to understand how deformation gestures are classified and used in order to test our prototype effectively. We made an effort to follow naming conventions used in previous research when designing experiments and explaining input requirements to participants (i.e. what “bend up” and “bend down” represent). We also considered the uses of gestures in applications other than games in order to better map the actions found in our games to the correct gestures.

2.2. Deformation Interactions With Games

Most research regarding deformation gestures tends to focus on performance-driven applications such as map navigation (Lahey et al., 2011), document navigation (Watanabe et al., 2008), and text input (Schwesig et al., 2004). Users describe games in two ways: (1) how good

they are at the game, and (2) how much they enjoy the game. Researchers have not thoroughly explored these topics within the subject of games that use deformation interaction.

Cobra (Ye & Khalid, 2010) is an all-in-one deformable handheld gaming system that consists of a flexible display board and a portable shoulder-bag porting a power source and pico-projector capable of tracking the flexible display. Cobra makes use of bend sensors attached to the display board in order to detect bending. They claim that gestures are dependent on the game being tested as different actions in-game requires different methods of input. We assume that the best control mappings for games are when the input gestures represent the in-game actions metaphorically, which was confirmed by Jessica Lo in her work with Bendy (2013). The researchers did no formal testing with Cobra but concluded that their device and deformable gestures in their entire would change how games are made. They did not, however, consider bendable input's use with current games.

Unlike with Cobra, Lo et al. (2013) looked into deformation input with existing games with their bendable prototype, Bendy. In their first experiment, participants were given a set of nine common tasks found in games and were asked to create gestures that would be mapped to these tasks. Each task had a subset of related tasks that also had to be mapped. For example, the shooting task had the subset tasks of triggering the shot, moving the shooter left, and moving the shooter right. Results revealed that users, for the most part, agreed on gesture mappings and commonly had an opposing event mindset when pairing to opposing gestures (i.e. bending one side would move one direction and bending the other side would move the other direction). They also noticed that participants held the device many different ways. In their second study, participants played six pre-existing games using control schemes that were designed based on the results of their first experiment. They found that participants were having positive reactions to

playing games using gesture input, but the inconsistencies in how users held the device led to some issues with participants needed to reposition their hands.

Nguyen et al. worked on two deformable prototypes, BendID (V. P. Nguyen, Yoon, Verma, & Ramani, 2014) and Softii (V. Nguyen, Kumar, Yoon, Verma, & Ramani, 2015). For BendID they created a prototype using conductive foam and an array of pressure sensors used to distinguish between various forms of deformation input. They state that they tested BendID using a 3D racing game but did not present any conclusive data regarding their informal experiments. The Softii prototype was made the same way and was also informally tested using a 3D racing game and a flight controlling game but again, no data was presented. Rendl et al. (2014) created a transparent flexible film that they believe could be used in conjunction with games requiring precision with multiple degrees of freedom. They suggest a variety of game mechanics that could easily be mapped to both discrete and continuous gestures already built into FlexSense. They did not, however, test FlexSense with any games.

Daliri and Girouard (2016) explored visual guides used to direct participants through applications on flexible display. They had participants perform an image docking task and a game. Users controlled a character who navigated through a grid collecting and eating various fruits. They discover that visual feedback did prove to be helpful when playing a game on a flexible display. Ahmaniemi et al. (2014) asked participants what applications they thought bends would work best with. 100% of participants suggested games. They specifically suggested games where you control speed, follow a track, or drop bombs. They also suggested specific games such as Angry Birds (Rovio Entertainment, n.d.) and Tetris (Pajitnov & Pokhilko, 1984).

We test our prototype with simple games based on suggestions found in this body of research. For our research, we stuck to existing games as they did in most of the related work in

order to determine which game mechanics map best to the deformation input. No prior work combined buttons with gesture input as we suggest it. BendID (V. P. Nguyen et al., 2014) and Softii (V. Nguyen et al., 2015) were the only prototypes to use an external display, however they did not complete any formal studies using games which we plan to conduct using our prototype.

2.3. Deformation Interactions With Other Input

We are interested in looking at combining deformation input with button input in our research to create a hybrid interaction technique. Researchers have combined deformation input with other types of input in the past in order to seek out their own hybrid interaction techniques. Kildal et al. (2013) developed a flexible prototype that also had touch capabilities on the front and back of the device. They used this prototype to test two hybrid interaction techniques against a basic touch technique in an image docking application. They made many observations but most notably they state that these hybrid techniques feel more intuitive than touch on its own and they definitely demonstrate potential once users are familiar with how the interaction works.

FlexView (Burstyn, Banerjee, & Vertegaal, 2012) is another example of combining touch and deformation interactions. Burstyn et al. used a deformable mobile device with touch screen in order to test z-axis navigation within mobile applications. Again, the deformation interaction, coupled with touch, demonstrated promise and allowed for increased performance in pan-and-zoom tasks. They also observed that interaction techniques that used absolute mapping over relative mapping (similar to the rate control against position control described by Strohmeier et al. (2016) in their ReFlex paper) performed much better. Rendl et al. (2014) created FlexSense, a transparent deformable surface, and demonstrated how it could be used as a deformable tablet cover to make use of both deformation gestures and touch. They believe that the flexible film could be used as a lens that, when folded back, could reveal new information, apply photo

effects, navigate in z-depth, or be used to create paper-like animation. Md Riyadh looked into tapping interaction done by the thumb of the holding hand of a device (Riyadh, 2014). They compared a rigid device to a flexible device and found that the accuracy of tapping interactions remained mostly constant between devices.

Konieczny et al. (2005) combined a flexible display with world space positioning in order to create an interactive scanning application that could be used to scan 3D models inside and out. Medical professionals who had a chance to test the device agreed that the device's ability to bend was desirable as it could better follow actual anatomical structures. Steimle et al. (2013) created a similar prototype they name Flexpad. Again, this prototype resembled a piece of paper that was trackable in three dimensions. They tested this prototype with three application examples: (1) making cuts in volumetric images, (2) controller virtual paper-like characters, and (3) slicing through time in videos. They found that users are able to mimic specific bends and folds with high precision using a prototype made of shape-retaining materials and a prototype without shape-retaining properties. Paddle, a device created by Ramakers et al. (2014), uses a Rubik's magic puzzle combined with touch input to demonstrate various applications that make use of physical controls. Their results suggested that the physical form of the device provided benefits and was preferred over touch input when completing leafing and peaking tasks. They believe that the physical aspect of the device allows for better use of motor skills, spatial memory, and comprehension of content. Tarun et al. (2011) created Snaplet, a flexible touch display that changes applications based on its current form. The device has three uses each defined by a different shape: (1) when the device is wrapped around the user's wrist it acts as a watch, (2) when the device is flat it acts as a PDA and (3) when held in a concave position it acts as a

phone. Although no testing was performed, they demonstrated the potential deformable interfaces can have regarding physical and contextual affordances and improved ergonomics.

Our prototype uses bend sensors alongside standard button input. We test multiple combined-interaction control schemes against basic button input in multiple games in order to accurately determine any objective or subjective differences. No prior work combined gestures with button input for any applications including games.

2.4. Novel and Natural Game Interaction

Devices such as Arduino (Arduino, 2016) and MaKey (JoyLabz LLC, 2016) are making it easier than ever to develop new types of interactions for video games. Bend sensors are a sample of the types of sensors available to integrate into the video-game experience.

Commercially, we have seen devices such as the Wiimote (Nintendo, 2006), the Guitar Hero guitar controller (Harmonix, 2005), and Microsoft's Kinect (Microsoft, 2010), which all represent novel methods of video game interaction. Another example is the Leap Motion (Leap Motion Inc, 2016) that performs accurate hand and finger tracking allowing users to precisely control their computer using their hands and fingers. Academically, most research focuses on input methods that can no longer be considered novel; touch, for example. Researchers such as Villar et al. (2007), Ionescu et al. (2011), and Smith (2013) are attempting something unique when tackling novel game interaction such as creating adaptable controllers, developing games for controller hybrids, and creating controllers that resemble the main character of the game. Naturalness is commonly associated to a positive user experience (V. P. Nguyen et al., 2014). Wigdor and Wixon (2011) define natural as a descriptor "we use to describe a property that is external to the product itself." They continue by explaining that "we see natural as referring to the way users interact with and feel about the product, or more precisely, what they do and how

they feel while they are using it.” Skalski et al. (2011) separate natural mapping into four distinct categories: directional, kinetic, incomplete tangible, and realistic tangible natural mapping. Through experimentation, they determine that natural mapping of a video game controller led to higher spatial awareness and enjoyment when playing games. Most novel controllers make use of one of these mapping strategies in order to remain intuitive and enjoyable.

There are some researchers who seek to create kits that allow gamers to either create their own controller or adapt current controllers to work better with specific games. Villar et al. (2007) created the VoodooIO gaming kit which is built to be adaptable based on the user’s preferences. VoodooIO presents users with a set of input components such as buttons and joysticks that they can connect and layout to fit their preference. They describe a cockpit setup that could be naturally mapped to a flight simulator game. Initial experiment results show that users are comfortable adapting their input device to their preference. Chatham et al. (2013) present a low level set of programming tools with their concept, Unojoy!. Their idea is to create an open-source library that will help in the rapid-prototyping of Arduino based video game controllers.

Some novel controllers attempt to combine traditional input in new ways or with sensors that are new and unique. Ionescu et al. (2011) created a system that uses a physical game controller alongside gestures captured by a 3D camera. They tested their system with a first person and third-person action game where the player needed to defeat enemies using a weapon (the physical controller) and magic (the gestures). Results reveal that users found the interactions natural and immersive as the two types of input provided them with the familiarity of the standard controller combined with the freedom of the hand-movement gestures. Squidge (Smith, 2013) is a controller modelled after the main character of the game it is used for. This controller

has many built-in sensors and was built to try and keep the narrative, mechanics, and interface of the game closely integrated. Although no formal experiments were conducted, they did manage to create a game based on the device that kept the user close to Squidge. Jones et al. (2014) did something similar using a device known as the Sphero. The Sphero is a spherical device that has the ability to move in any direction at various speeds on its own. They created a game that used Spheros to control other Spheros in attempt to take advantage of the physical properties of the controller in order to make a more immersive game. Again, without conducting any formal experiments, they concluded that this experience was more natural and enjoyable than the typical Sphero game which is controlled by a smartphone. Other unique uses of sensors and technology with games include a system that projects games onto peoples' bodies (Cassinelli, Angeseva, Watanabe, Frasca, & Ishikawa, 2012), the use of a Rubik's snake to control a samurai sword's shape (Inoue, 2015), a cylindrical motion detecting wand made with two flexible organic light-emitting diode screens (Priyadarshana & Porter, 2016), and a collaborative multiplayer multi-device game of Asteroids that makes use of multiple tablets and fiducial markers in order to control the ships and in-game actions (Garcia-Sanjuan, Jaen, Catala, & Fitzpatrick, 2015).

One unique controller that has similar functionality to that we intended to create is Namco's neGcon controller ("Namco neGcon Controller," 1995). This controller was built for a game called Ridge Racer and was made of two rigid halves, connected by a dowel, that could be twisted relative to each other. The twisting was used as the input for turning the car in the game. This controller only worked with Ridge Racer and did not allow for bends or any freedom with the twisting (twisting had to be done on the axis of the dowel). The neGcon controller can be seen in Figure 1.



Figure 1. Namco's neGcon Controller (Namco Bandai Games, 2015)

Our prototype is designed to be novel as it combined traditional video game input methods with bend sensors and deformation gestures. Although technology such as the Kinect and Leap Motion are able to detect similar hand gestures, we wanted to study the physical gestures alongside buttons, another physical method of input. We attempted to map our control schemes as naturally as possible in order to maximize fun and intuitiveness based on suggestions and results in this body of research. Our intentions are to create a controller that not only works objectively, but is subjectively fun to use and encapsulates the users in the games.

Chapter 3. Prototype

We sought to create a video game controller to test flexible input combined with binary input for simple video games. We based our prototype on the Nintendo Entertainment System (NES) game pad (as seen in Figure 2), which is a game pad used to play the simple video games we were seeking to test. In short, we added a flexible bridge in the middle of the game pad, between the directional buttons and the action buttons. We used the prototype to test PC ports of NES games as well as a PC-based racing game.



Figure 2. Nintendo Entertainment System Controller

3.1. Interaction Language

Our controller has ten inputs: four deformations and six buttons. Although prior work has discussed the possibility of over thirty gestures (Lee et al., 2010) we wanted a smaller number in order to not overwhelm our participants. The four bend/twist gestures are as follows: (1) bend up (2) bend down (3) twist left, and (4) twist right. Examples of the different input states can be found in Figure 3. We define bend up as the bridge arching upwards, and the back of the panels

being bent towards each other. We define bend down as the bridge arching downwards, and the front of the panels being bent towards each other. These gesture definitions mimic those defined by Kildal et al. (2012). We define twist left as bringing the top of the right panel towards the user and the top of the left panel away from the user. Finally, we define twist right as rotating the top of the right panel away from the user and the top of the left panel towards the user. Twists were defined that way to simulate how people activate automobile turn-signals: rotating your left hand away from your body (flick down) is used to signal left whereas rotating your left hand towards your body (flick up) is used to signal right.

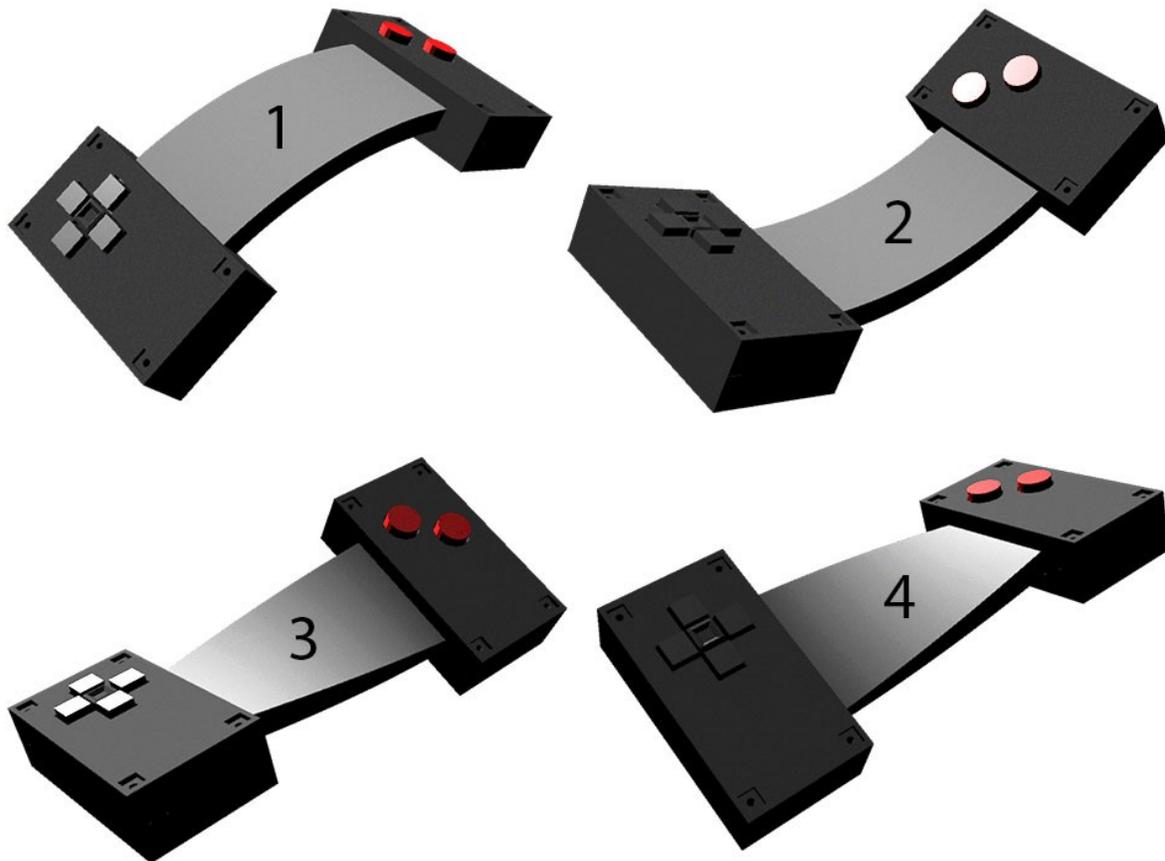


Figure 3. The Four Possible Controller Gestures; (1) Bend Up (2) Bend Down (3) Twist Left (4) Twist Right

Our controller has six buttons, four on the left panel and two on the right panel. We refer to the four buttons found on the left panel as: up, down, left, and right. We refer to the two

buttons on the right panel as action 1 (left) and action 2 (right). In the original NES controller there were two additional buttons that represent the “start” and “select” buttons, but we decided to remove them to give us more internal wiring space for the directional pad.

3.2. Hardware

We built a handheld game controller with rigid side panels connected by a flexible bridge that is able to bend and twist. We designed the controller to be held with both hands: it measures 162 x 75 x 21 mm. We modified the original NES game pad design and dimensions slightly (original NES dimensions: 125 x 55 x 17 mm) to implement the flexibility bridge, and modified the button positions after having tested it with multiple hand sizes. Figure 4 displays a schema of the controller from a top-down perspective, while Figure 5 displays the bottom view. We 3D printed the side-panels (40 x 75 x 21 mm each) using polylactic acid (PLA) filament which produces a rigid plastic end-product. The flexible bridge, measuring 82 x 43 x 6 mm, is made of two 2-inch bend sensors fastened on the rear side of a foam board cutout.

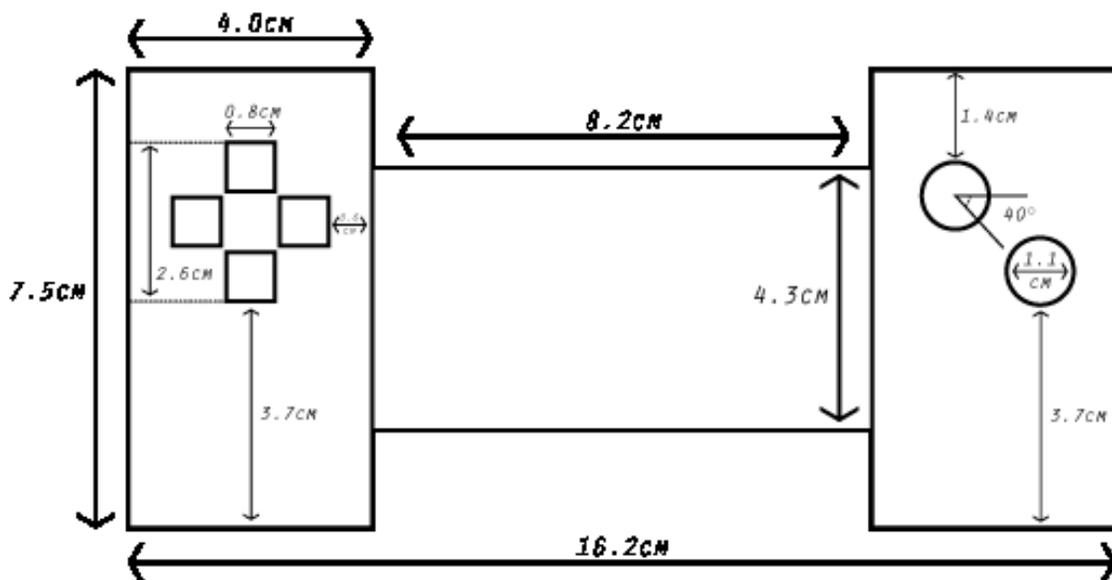


Figure 4. Controller Dimensions Top-Down

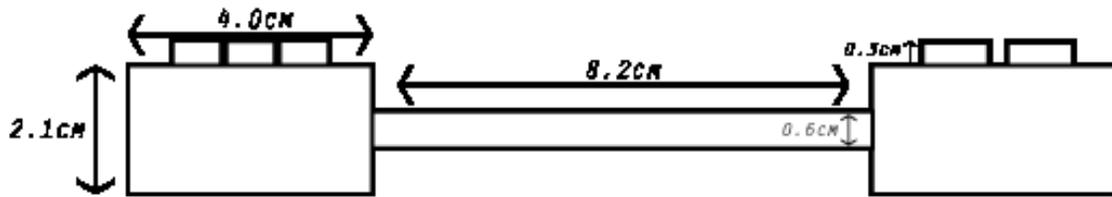


Figure 5. Controller Dimensions Front View

We placed the two buttons on the right panel near the top at a 40° angle from one another, to simulate the angled buttons found on more modern game controllers, which fit more naturally under a user’s thumb when holding the device.

When designing the flex bridge, we tested many materials such as plastic, rubber, and foam, but each material presented us with its own set of problems. It was difficult to twist the plastic materials and they tended to deform and warp quite easily. The rubber we tested was too thin and when layered made it difficult to both bend and twist in either direction. Foam was our best option as it was malleable enough to bend and twist in all directions and could retain its shape fairly well, even after excessive use.

The internal bend sensors overlap diagonally in the centre of the flexible bridge to accurately distinguish between our four gesture-based input methods: (1) bend up (2) bend down (3) twist left, and (4) twist right. For the first prototype, we fastened these sensors from their base and placed them in plastic sleeves to give them some ability to slide. The front and back view of the final prototype used in experiment one can be found in Figure 6 and Figure 7. For the second study, we modified the prototype so that the sensors no longer needed to be in plastic sleeves by securing them in the centre using a piece of tape lined with a piece of non-stick plastic. This was necessary because when bending and twisting, the foam stretches, but the bend sensors could not. The front and back view of the final prototype used in experiment two can be found in Figure 8 and Figure 9.



Figure 6. Front side of Prototype for Study 1



Figure 7. Backside of Prototype for Study 1

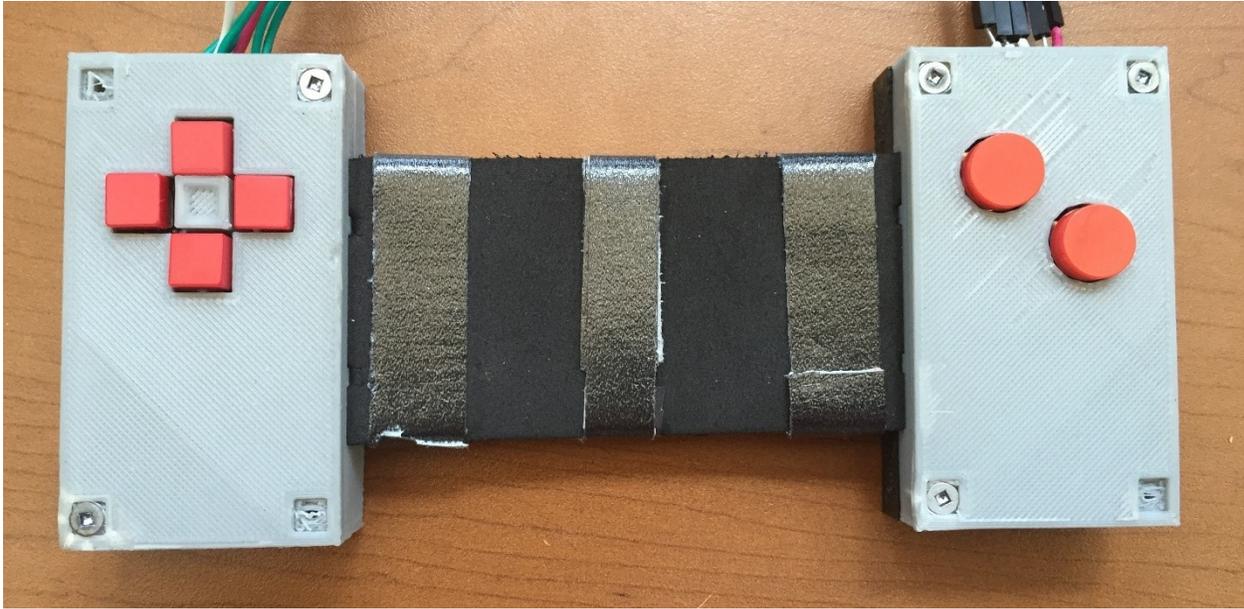


Figure 8. Front side of Prototype for Study 2

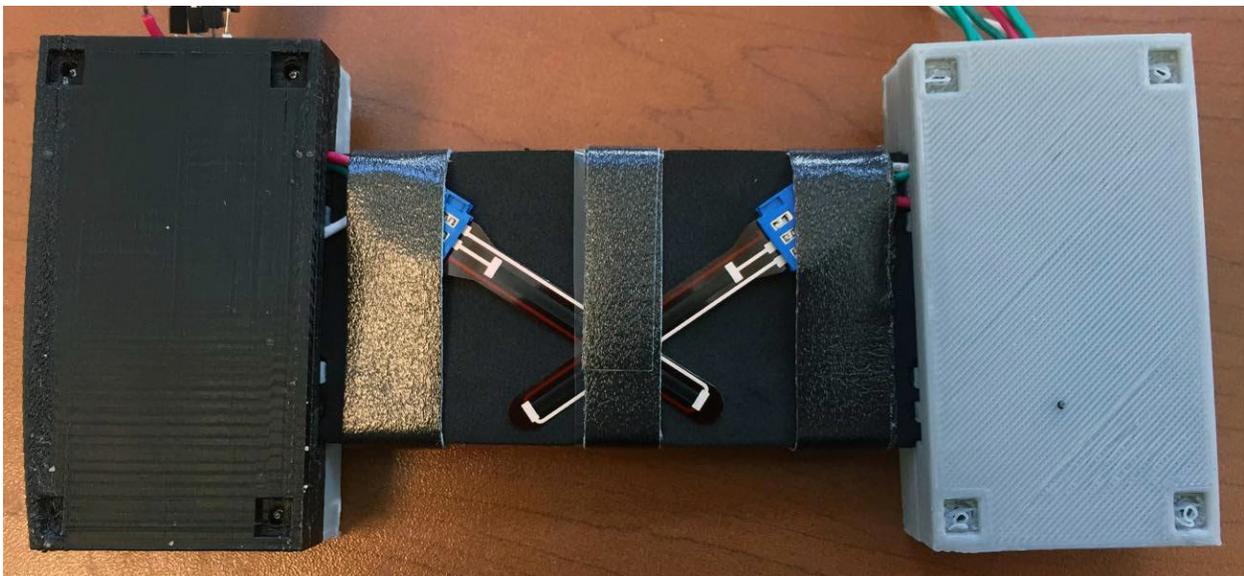


Figure 9. Backside of Prototype for Study 2

All wires emerge from the top of either panel. Originally, we had all wiring from the right panel crossing through the flex bridge and emerging out of the left panel, however, due to the nature of bend and twist input, these wires were coming loose and breaking too often. These emerging wires are connected to two 48 cm long ribbon cables, which are then connected to an

external Arduino Leonardo. The Arduino Leonardo is connected to a MacBook Pro via USB.

The Arduino has one additional button used to collect data and calibrate the controller which we will refer to as the calibration button.

3.3. Software

We used Arduino 1.6.7 to interpret the input data from the controller. Each button has an up and a down state so that the buttons can be held down for continuous input. We analyze the raw bend sensor data in order to determine which gesture is being input. The raw bend sensor data reads between 0 and 1024, 512 being the hypothetical rest position. We use a sensitivity threshold ($c=80$) to minimize accidental input. The sensitivity threshold (c) is the amount of bending, from the rest position, required for the program to recognize it as being bent. We also implemented a calibration system that sets the rest positions of the bend sensors based on their average input values over a period of ten frames at 66.67 Hz. In experiment 1, calibration occurs on upload and when the user presses the calibration button on the Arduino. In experiment 2, calibration occurs before every race. This process is quick as it lasts only ten frames. In experiment 1, “bend Up” is triggered when both sensors read higher than the rest position plus the sensitivity threshold. “Bend Down” is triggered when both sensors read lower than the rest position minus the sensitivity threshold. “Twist Left” is triggered when the left sensor reads higher than the rest position plus the sensitivity threshold and the right sensors reads below its rest position. “Twist Right” is triggered when the right sensor reads higher than the rest position plus the sensitivity threshold and the left sensor reads below its rest position. In experiment 2 gestures are triggered the same way but their data is read continuously as opposed to discretely. In experiment 1, gestures, like buttons, also have up and down input states. This means that when the user triggers a gesture, it cannot be triggered again until the user restores the controller to its

rest position. The sensors must only be back in their rest position for one frame in order for them to be restored.

In experiment 1, participants will be playing three games ported to PC from the NES. The NES ports are controlled using the keyboard. Our prototype is connected to an Arduino Leonardo which is connected to a MacBook Pro. We make use of the Arduino virtual keyboard library to simulate key presses with button presses and gesture input in order to play the game using our prototype. We set up our program to work with an array of control schemes, which are sets of mappings between the input and resulting virtual key presses. We can switch these schemes by changing a single integer in the code and re-uploading to the Arduino. We needed this ease of use to play various games while restricting the participants input to the specified control scheme.

We used a small Processing program to track input performance data. This program runs in the background and simply picks up the key pressed simulated through the Arduino's virtual keyboard when a key press is triggered. We save the data to a uniquely identified .txt file at the end of each run. We track every time a button or gesture input has started and ended. We also track the current participant ID, the current game, the current control scheme, and the keyboard input translation. The apparatus for experiment 1 can be found in Figure 10.

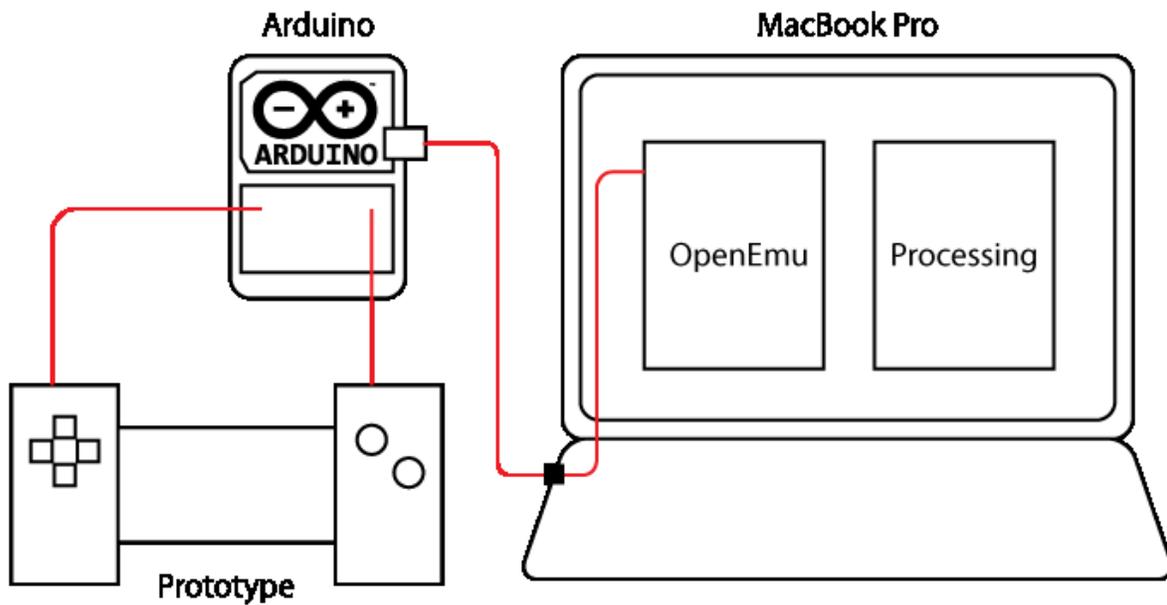


Figure 10. Apparatus, Experiment 1

In experiment 2, participants play a racing game developed and run in Unity3D. The Arduino is used to read sensor data and write that data to the serial port. Unity then reads the serial port and parses the sensor data for use within its scripts. Bends and twists are triggered the same way as they are in experiment 1 except through Unity as opposed to through the Arduino. All data collection for experiment 2 is done in Unity and is written to a .txt file every time a race is completed. We developed an interface within the game in order to handle control scheme switching and track selection. The apparatus for experiment 2 can be found in .

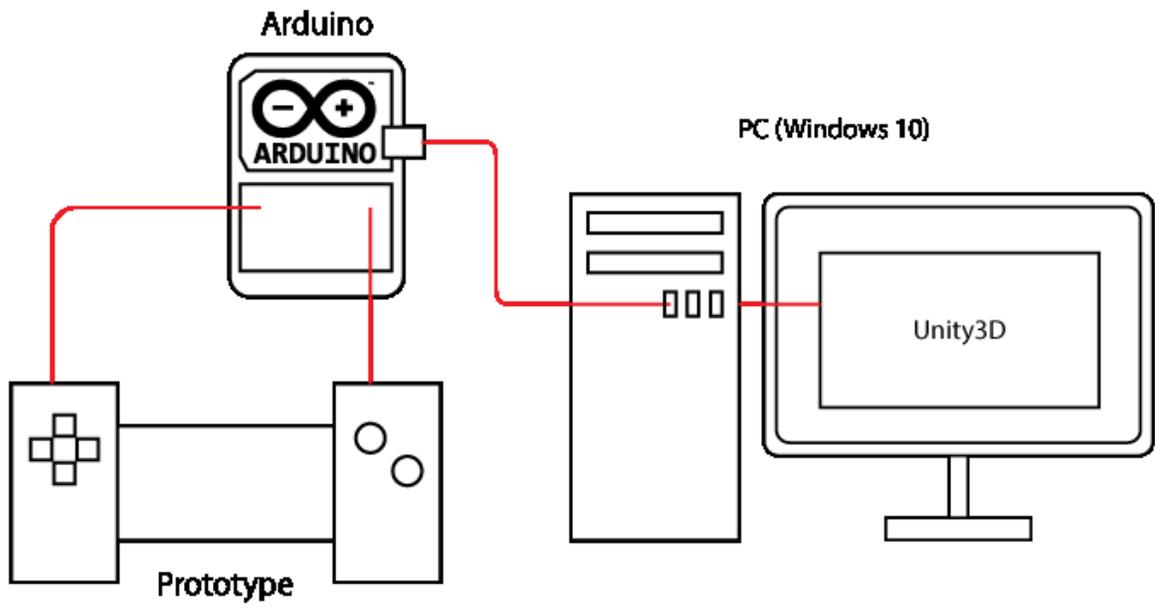


Figure 11. Apparatus, Experiment 2

Chapter 4. Experiment 1: Flexible Input Methods for Classic Video Games

4.1. Overview

Our primary research goal is to determine if flexible input, combined with binary input, is a satisfying method of input when playing simple video games using our custom built controller. Our secondary research goal is to determine the differences between uses for bend controls in specific games. We could map buttons and gestures to an endless number of in-game actions given the variety and number of games in the market. Genres are a good way of classifying games based on their in-game mechanics and gameplay. Genres are described as groups of games that share similar properties and game mechanics. Generally, there is a commonly accepted group of games genres that ranges between 12 and 20 unique categories including shooters, role-playing games, rhythm games, and action games. There is no official list but those examples alongside a few others are the genres found most frequently in today's media. Games, for the most part, are very unique and no two games control the same leaving an uncountable number of possible input-to-action combinations. For example, pressing a button could make your character jump in one game, but in another, the same button being pressed causes an explosion. A bend up in one game could move your character up, but in another could activate a magic ability. Genres, as mentioned above, help to maintain some expectations and standards between both games and how they control. For organizational purposes, we created two types of mapping between bend gestures and game actions: one that focuses solely on common in-game actions (such as jumping or causing an explosion), and the other on in-game navigation (moving objects in 2D/3D space). This allowed us to generalize the control schemes to give the participants a better immediate idea of what the gestures would be mapped to in each situation.

We also chose games from different genres in order to determine which types of actions and navigation would map the best to bend and twist input.

4.2. Games

We selected games based on three criteria: our ability to isolate the action and the navigation within the games, the ability to measure performance objectively, and a diversity within gameplay. The first criterion allowed us to evaluate our second goal by creating control schemes based on both action and navigation mappings. We needed the games to have specific and similar performance measurements so that we could measure performance consistently and objectively across games. We were also looking for diversity within gameplay to ensure that the experience with each game was mutually exclusive (learnability did not transfer from one game to another within the duration of the experiment). We selected three games for our experiment: Donkey Kong (Miyamoto, 1981), Mike Tyson's Punch-Out (Genyo, 1987), and Tetris (Pajitnov & Pokhilko, 1984). All three games are ROMS played using an NES emulator on the MacBook Pro.

The first game, Donkey Kong (Figure 12) is an arcade platformer released on the Nintendo Entertainment System (NES) in 1981. In Donkey Kong, the user plays as Mario and has to rescue the princess from Donkey Kong himself, who reigns over each level tossing barrels at Mario. Mario can jump, move left and right, and climb ladders. To rescue the princess, Mario must reach the top platform while avoiding incoming barrels and fire. The game distributes points to users for jumping over barrels, destroying barrels with the hammer, and reaching the end of the level. Mario can die three times before a game over.

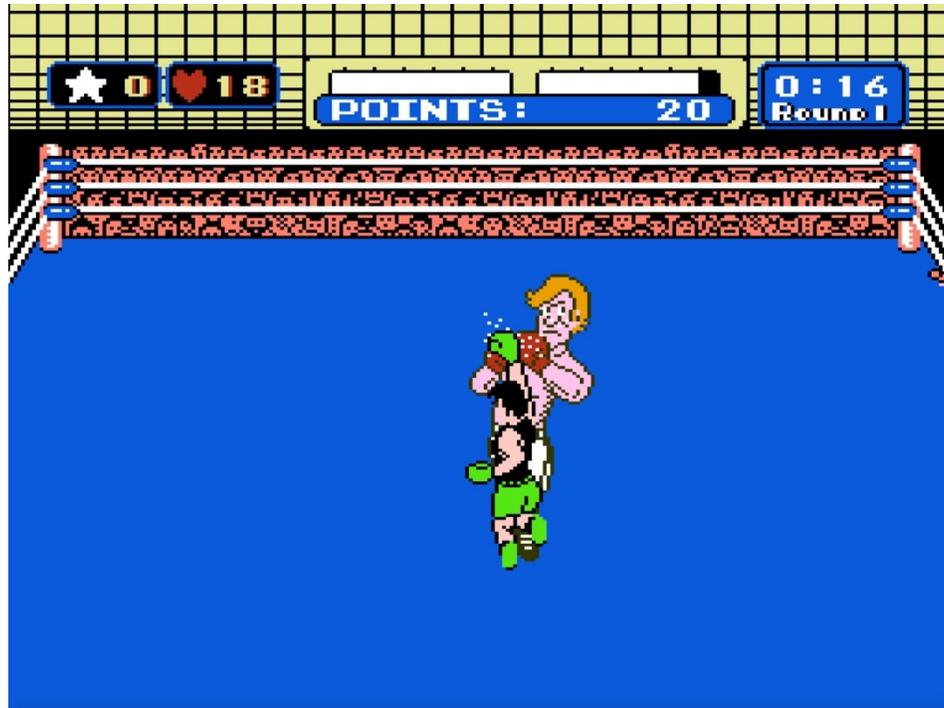


Figure 13. Mike Tyson's Punch-Out (Genyo, 1987)

Finally, Tetris (Figure 14) is a tile matching puzzle game originally released for the NES in 1989. In Tetris, the player controls puzzle pieces that are coming down from the top of the screen. Users can move and rotate the puzzle pieces as they fall towards the bottom of the screen. The player must try and create horizontal lines across the playing grid to clear the blocks. Players score points by placing pieces strategically, clearing lines, and completing Tetrises (clearing four lines at once). The player loses if the pieces reach the top of the screen.

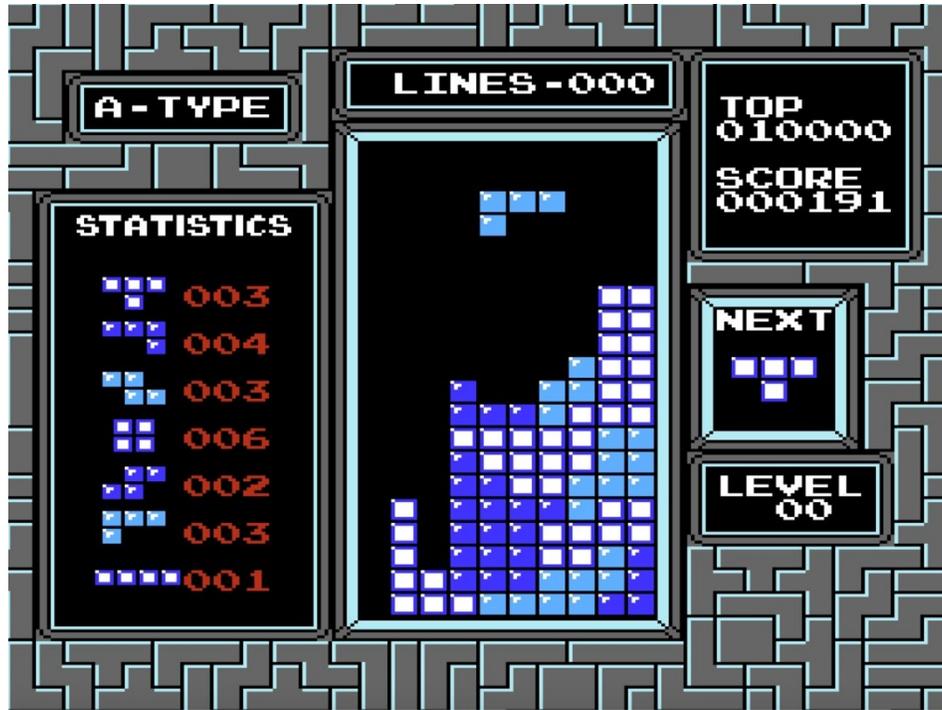


Figure 14. Tetris (Pajitnov & Pokhilko, 1984)

We are using OpenEmu (OpenEmu, 2013) to run read-only memory (ROM) versions of our three test games: Donkey Kong, Mike Tyson's Punch-Out, and Tetris (Pajitnov & Pokhilko, 1984). All three games are the original NES versions. We are able to change the controls from within this program so that they fit with our control schemes. Both the physical keyboard and virtual keyboard found inside the Arduino can control the games.

4.3. Control Schemes

We tested three classifications of control schemes that we modified slightly for each game. The first control scheme is only binary input, and is representative of the game's original control scheme. Binary is the name of the scheme that only makes use of button input. The other two schemes combine binary and gesture input methods. We differentiated these two control schemes by the type of in-game mechanic controlled by the gesture input: one scheme where we

used gesture to control action input and the other where we used bend/twist to control navigation input. We separate game tasks like so in order to determine which types of in-game tasks map best to gestures. We mapped all other necessary inputs to binary controls. We disabled all input with no direct action in-game. For Table 1., Table 2., and Table 3, the following abbreviations apply: twist (T), bend (B), action (A), d-pad (D), up (u), down (d), left (l), and right (r). Multiple input in a cell separated by a “/” means that either input method will work to trigger that action. Multiple input in a cell separated by a “+” means that users must activate both inputs to trigger that action. We did not implement a control scheme that only made use of gestures as all three games required more than four unique inputs (where we could only offer four gesture inputs). Visual representations of all nine control schemes can be found in Figure 15.

| | UP | DOWN | LEFT | RIGHT | JUMP |
|-------------------|-----------|-------------|-------------|--------------|-------------|
| BINARY | Du | Dd | Dl | Dr | A1/A2 |
| ACTION | Du | Dd | Dl | Dr | Bu/Bd/Tl/Tr |
| NAVIGATION | Bu | Bd | Tl | Tr | A1/A2 |

Table 1. Donkey Kong Control Schemes

| | RIGHT | LEFT | DUCK | PUNCH | PUNCH | UPPERCUT | UPPERCUT |
|-------------------|--------------|-------------|-------------|--------------|--------------|-----------------|-----------------|
| | | | | LEFT | RIGHT | LEFT | RIGHT |
| BINARY | Dr | Dl | Dd | A1 | A2 | Du+A1 | Du+A2 |
| ACTION | Dr | Dl | Dd | Tl | Tr | Tl + (A1/A2/Du) | Tr + (A1/A2/Du) |
| NAVIGATION | Tr | Tl | Bd | A1 | A2 | Bu+A1 | Bu+A2 |

Table 2. Mike Tyson’s Punch-Out Control Schemes

| | DOWN | LEFT | RIGHT | ROTATE R | ROTATE L |
|-------------------|-------|------|-------|----------|----------|
| BINARY | Dd | DI | Dr | A1 | A2 |
| ACTION | Dd | DI | Dr | Tr | Tl |
| NAVIGATION | Bd/Bu | Tl | Tr | A1 | A2 |

Table 3. Tetris Control Schemes

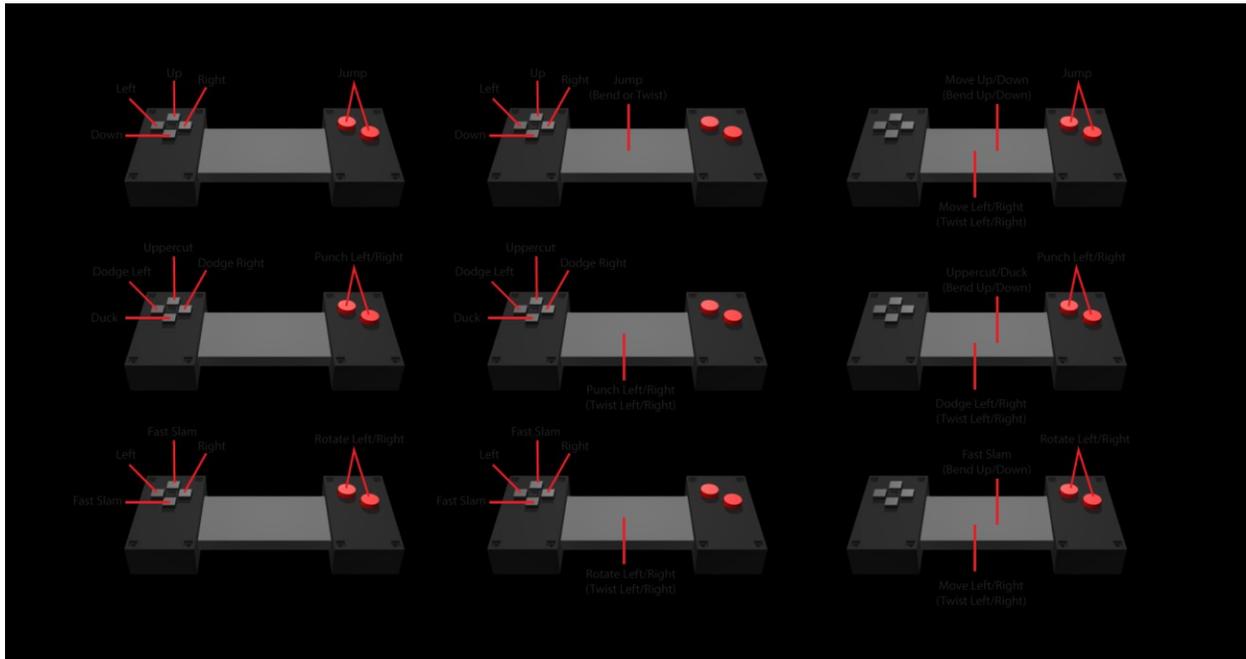


Figure 15. Control Schemes for Experiment 1

4.4. Methods

We tested each of the three control schemes with each of the three games for a total of nine conditions for each participant. Each condition consisted of two trials that lasted two minutes each. Each session took approximately 80 minutes. We presented the games in a predetermined pseudo-random order to ensure that each game was played in every position an equal number of times. Within each game, participants played the three control schemes also in a

pseudo-random order. We explained each game to participants, and offered them the opportunity to play the game using the keyboard before the recorded trials began, to make sure they were comfortable playing the game, minimizing the measurement of the learnability of the game itself.

We measured objective performance in the forms of score, max level, and deaths for each game. In the case of Tetris, we mapped the max level metric to the number of lines cleared, while it was mapped to the furthest round the player made it to in Mike Tyson's Punch-Out. In Donkey Kong a level was considered complete when the player managed to defeat Donkey Kong. We measured subjective cognitive load using the NASA task load index (TLX) questionnaire and the user experience (UX) using the User Experience Questionnaire (UEQ) (Laugwitz, Held, & Schrepp, 2008). The NASA-TLX is a test used to determine subjective workload of a system over six unique scales including mental demand, physical demand, temporal demand, performance, effort, and frustration. The UEQ is a quick assessment tool used to evaluate an overall user experience with a product or system. It produces results over six categories (attractiveness, perspicuity, efficiency, dependability, stimulation, and novelty) that can be further grouped into three categories: (1) attractiveness, (2) pragmatic quality, and (3) hedonic quality. Due to an error creating our measurement survey, we measured the UEQ using a 5-point Likert scale instead of a 7-point. We presented both questionnaires after each game/scheme combination.

We asked participants to fill out a game survey after their completion of each game (meaning they have played the game with all three control schemes). The game survey asked participants to rank control schemes based on four criteria for each game: most fun, most comfortable, most frustrating, and most natural. We presented participants with a final questionnaire to collect ethnographic data and asked general questions about our flexible

controller. All questionnaires can be found in the appendices (9.2.1. Experiment 1). This work was approved by Carleton University's Research Ethics Board.

4.5. Hypotheses

Although it would be nice to see increased performance using gesture-based control schemes, we predict that the binary scheme will out-perform both the action and navigation schemes in terms of score, max level, and death count (H1). We also predict that the binary scheme will receive the lowest overall workload across all three games and score higher in perspicuity on the UEQ (H2). We believe this will be the case because binary control schemes are already familiar to most people who have experience playing video games. Users do not usually encounter or use gesture-based schemes in video games, so unless users have previous experience using deformation input (which most do not), users will perceive these schemes as more complicated and less secure, leading to a higher overall workload. The main reason for including a binary control scheme is to provide us with baseline scores to which we can compare the scores that the gesture schemes receive. We believe that our gesture-based schemes will not immediately out-score binary as this is an introductory study for our prototype using hybrid control schemes that have yet to be tested.

Our third hypothesis is that gesture-based schemes will be more fun and receive higher attractiveness and hedonic scores in the UEQ (H3). Deformation input, being relatively new, will be seen as novel to most participants and using novel input to do something inherently fun, such as playing video games, will only augment their stimulation levels and sense of enjoyment. We are only testing participants who are new to our prototype so further testing would need to be done in order to determine any immediate impact the novelty effect may have on our results.

Finally, we predict that the action scheme will perform better than the navigation scheme across all three games (H4) and be less demanding in terms of subjective workload (H5). We believe the action scheme will be less demanding as this scheme consistently requires less deformation input across all three games compared to the navigation scheme. We also predict that action will perform better over navigation due to its relative simplicity and comparatively smaller learning curve.

4.6. Participants

We had participants sign a consent form before beginning the experiment. This consent form can be found in the appendices in chapter nine (9.1.1. Experiment 1). Our 16 participants (3 female) were all right-handed (mean age of 22.56). 13 participants reported to play video games for more than one hour per week, with 4 playing for more than ten hours. 14 participants had experience playing video games with novel controllers, such as the Guitar Hero guitar, the Wiimote, or Microsoft's Kinect. On a scale of one to five (one being no experience and five very experienced), the average familiarity score for Donkey Kong was 2.94 (SD = 1.53), for Punch-Out was 1.56 (SD = 0.81), and for Tetris was 4.00 (SD = 1.03). Two participants had experience using flexible input in the past. We presented participants with a \$10 Tim Horton's gift card as compensation.

4.7. Results & Analysis

We analyzed the games individually, as their game play and resulting user experiences were too different for us to consider them as one whole.

4.7.1. Performance

We measured score, max level achieved, and deaths for every participant for each game/scheme trial. Every participant played every game with every control scheme apart from one participant who did not play Tetris ($n = n - 1$) due to unforeseen time constraints. We only analyze the values for the second trial, to avoid measuring the initial learnability of the game/scheme combination. An overview of the performance statistics can be found in Table 4.

| Measurement | Game | Binary Mean (SD) | Action Mean (SD) | Navigation Mean (SD) | Stats Binary/Action Binary/Navigation Action/Navigation |
|-------------|-------------|---------------------|---------------------|-------------------------|--|
| Score | Donkey Kong | 5500 (4266) | 2581 (2479) | 1919 (2883) | $p = 0.032$ $p = 0.001$ not significant |
| | Punch-Out | 2403 (1217) | 1983 (1151) | 1544 (1345) | not significant $p = 0.017$ not significant |
| | Tetris | 1393 (1687) | 865 (1000) | 469 (468) | not significant $p = 0.049$ not significant |
| Max Level | Donkey Kong | 1.88 (0.89) | 1.44 (0.51) | 1.38 (0.72) | not significant $p = 0.019$ not significant |
| | Punch-Out | 2.19 (0.40) | 2.00 (0.37) | 2.13 (0.34) | not significant not significant not significant |
| | Tetris | 9.07 (6.10) | 7.00 (4.91) | 4.47 (4.22) | not significant $p = 0.003$ not significant |
| Deaths | Donkey Kong | 2.94 (1.34) | 3.75 (0.68) | 3.81 (1.42) | not significant $p = 0.017$ not significant |
| | Punch-Out | 0.38 (0.50) | 0.56 (0.73) | 0.50 (0.52) | not significant not significant not significant |
| | Tetris | 0 (0) | 0 (0) | 0.07 (0.26) | not significant not significant not significant |

Table 4. Overview of Performance Statistics for Experiment 1

4.7.1.1. Score

We performed a repeated measures ANOVA with a Greenhouse-Geisser correction on the score, and found significant differences between control schemes for Donkey Kong (F [1.726, 25.890] = 10.096), $p = 0.001$). We used post hoc tests using the Bonferroni correction to

investigate pairwise comparison. The binary scores were significantly higher than the action ($p = 0.032$) and navigation scores ($p = 0.001$).

We performed the same analysis for Mike Tyson’s Punch-Out, and found significant differences ($F [1.799, 26.969] = 5.397$), $p = 0.013$). Post hoc tests revealed significant differences between the binary and navigation scheme ($p = 0.017$).

We also found statistical significance for Tetris ($F [1.529, 21.410] = 3.930$), $p = 0.045$). Post hoc tests using the Bonferroni correction revealed that, similarly, the only two schemes that scored significantly different were binary and navigation ($p = 0.049$). Figure 16 presents the mean score for each combination. The black lines above the data represent which scores were significantly different from one another.

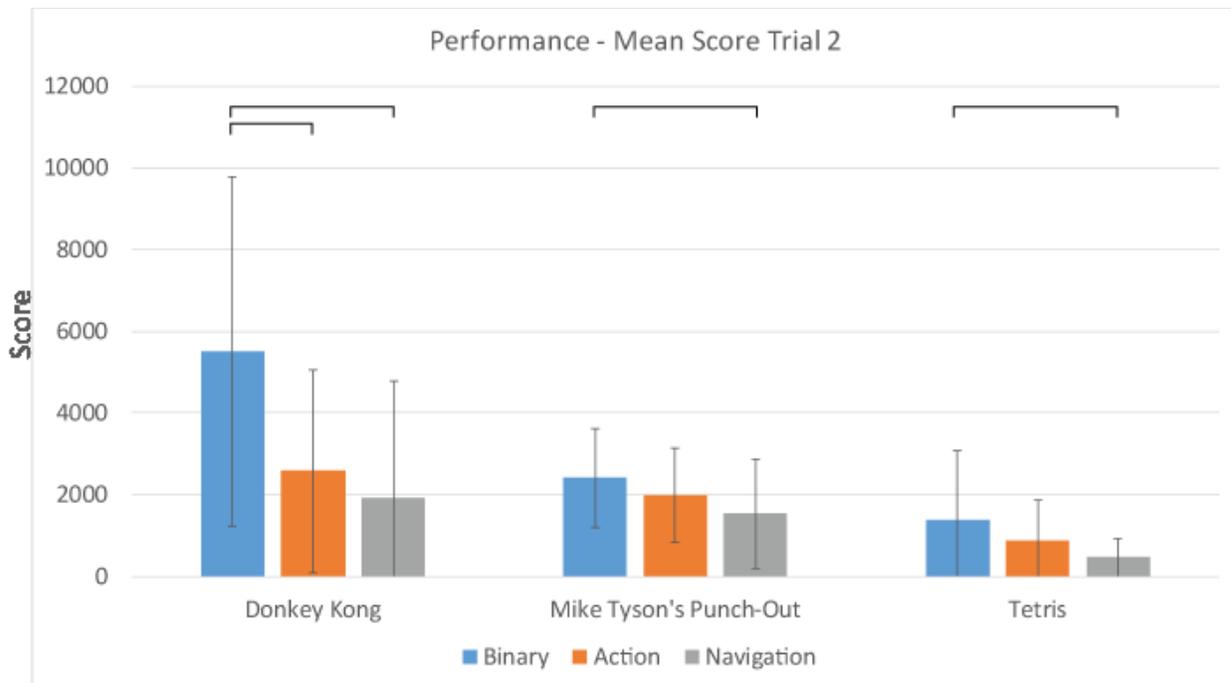


Figure 16. Performance Mean Scores with Standard Deviation and Significant Differences (black horizontal lines)

4.7.1.2. Max Level

We performed a repeated measures ANOVA with a Greenhouse-Geisser correction, which determined that the mean max-level achieved differed statistically significantly between control schemes for Donkey Kong ($F [1.609, 24.141] = 5.120$), $p = 0.019$). Post hoc tests using the Bonferroni correction revealed that the binary scheme achieved a significantly higher max-level mean when compared to the navigation scheme ($p = 0.005$).

The same test determined that the mean max-level differed statistically significantly between control schemes for Tetris ($F [1.594, 22.313] = 5.619$), $p = 0.015$). Post hoc tests using the Bonferroni correction revealed that the binary scheme reached the highest mean level but this was only significant when compared to the navigation control scheme ($p = 0.003$).

We did not find any significant differences between max level reached between control schemes in Punch-Out. Mean max levels can be found in Figure 17.

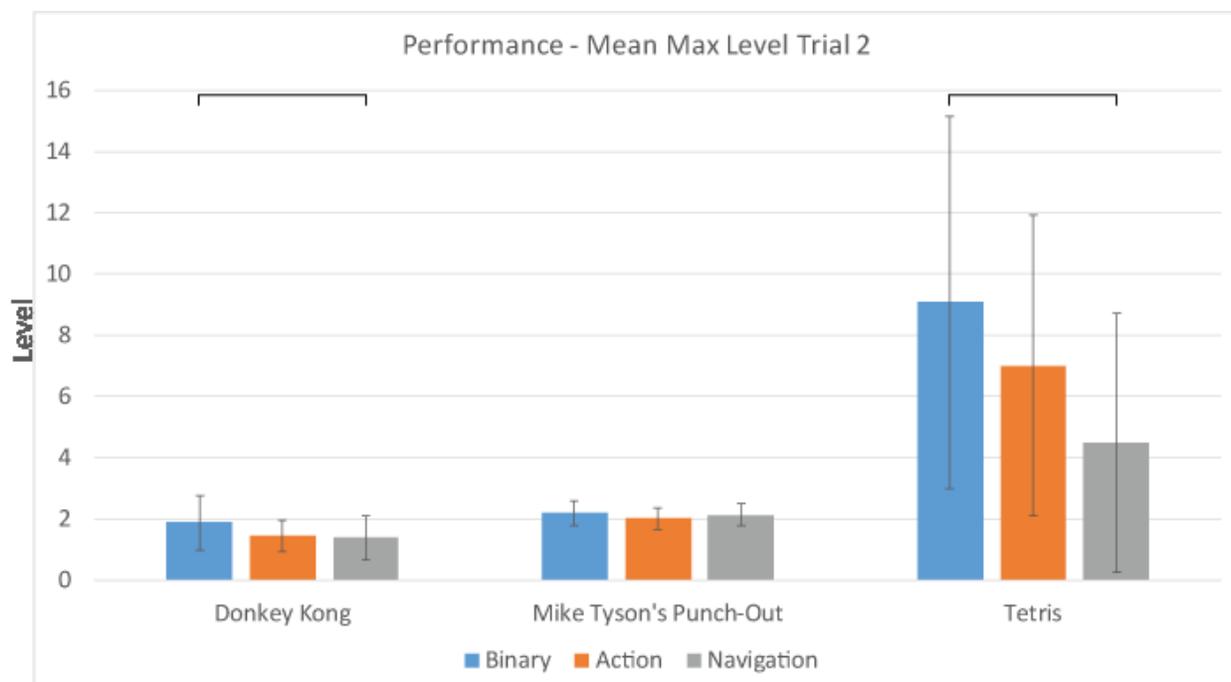


Figure 17. Performance Mean Max Level with Standard Deviation and Significant Differences

4.7.1.3. Deaths

We performed a repeated measures ANOVA with a Greenhouse-Geisser correction, which determined that the mean death count differed statistically significantly between control schemes for Donkey Kong ($F [1.764, 26.458] = 4.336$, $p = 0.027$). We found a significant difference in number of deaths when comparing binary to navigation ($p = 0.017$) with participants dying fewer times when using the binary control scheme.

The repeated measures ANOVA with a Greenhouse-Geisser correction test did not reveal any significant differences in deaths between control schemes for Punch-Out or Tetris. Mean death counts can be found in Figure 18.

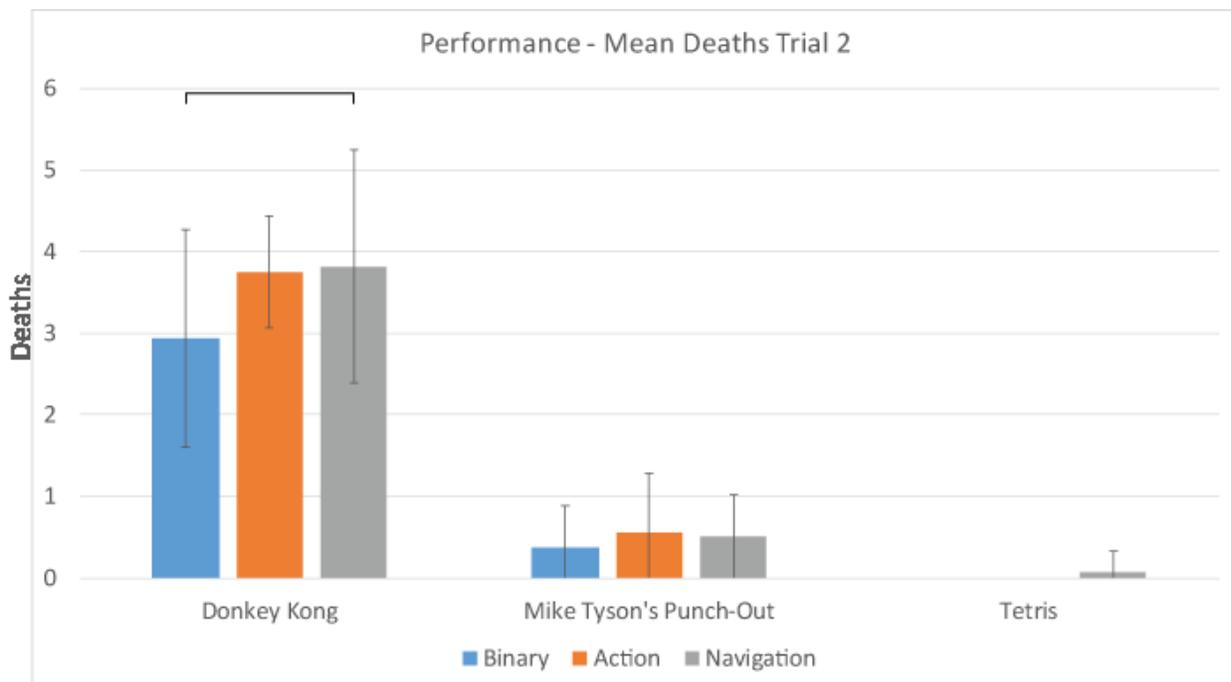


Figure 18. Performance Mean Death with Standard Deviation and Significant Differences

4.7.2. Game Experience

Participants were asked to rank the schemes in order from 1 to 3 based on the given question criteria. The x-axis for the ranking graphs represents total number of votes received.

The y-axis for the ranking graphs represents the rankings themselves, from 1 to 3. For example, in Figure 19, binary was ranked as number 1 five times, number 2 nine times and number 3 two times. These questions were asked supplementary to the UEQ as we wanted more detailed data regarding these four topics (fun, comfort, naturalness, and frustration). A summary of the game ranking statistics can be found in Table 5.

| Measurement | Game | Binary Mean (SD) | Action Mean (SD) | Navigation Mean (SD) | Stats Binary/Action Binary/Navigation Action/Navigation |
|--------------------|-------------|---------------------|---------------------|-------------------------|--|
| Fun | | | | | |
| | Donkey Kong | 1.81 (0.66) | 2.31 (0.87) | 1.88 (0.89) | not significant not significant not significant |
| | Punch-Out | 2.00 (0.73) | 1.88 (0.89) | 2.13 (0.89) | not significant not significant not significant |
| | Tetris | 1.93 (0.80) | 1.60 (0.74) | 2.47 (0.74) | not significant not significant not significant |
| Comfort | | | | | |
| | Donkey Kong | 1.25 (0.58) | 2.56 (0.73) | 2.19 (0.54) | ($Z = -2.911, p = 0.004$) ($Z = -2.982, p = 0.003$) not significant |
| | Punch-Out | 1.81 (0.83) | 2.00 (0.89) | 2.19 (0.75) | not significant not significant not significant |
| | Tetris | 1.47 (0.74) | 1.80 (0.68) | 2.73 (0.46) | ($Z = -3.040, p = 0.002$) not significant ($Z = -2.841, p = 0.005$) |
| Frustration | | | | | |
| | Donkey Kong | 2.94 (0.25) | 1.56 (0.63) | 1.50 (0.65) | ($Z = -3.380, p = 0.001$) ($Z = -3.624, p < 0.0005$) not significant |
| | Punch-Out | 2.63 (0.62) | 1.88 (0.81) | 1.50 (0.63) | not significant ($Z = -3.051, p = 0.002$) not significant |
| | Tetris | 2.93 (0.26) | 1.93 (0.46) | 1.13 (0.35) | ($Z = -3.260, p = 0.001$) ($Z = -3.626, p < 0.0005$) ($Z = -2.828, p = 0.005$) |
| Naturalness | | | | | |
| | Donkey Kong | 1.13 (0.32) | 2.44 (0.73) | 2.44 (0.51) | ($Z = -3.198, p = 0.001$) ($Z = -3.666, p < 0.0005$) not significant |
| | Punch-Out | 1.63 (0.72) | 2.06 (0.85) | 2.31 (0.79) | not significant not significant not significant |
| | Tetris | 1.20 (0.41) | 1.93 (0.59) | 2.87 (0.35) | ($Z = -2.399, p = 0.016$) ($Z = -3.542, p < 0.0005$) ($Z = -2.841, p = 0.005$) |

Table 5. Summary of Game Ranking Statistics for Experiment 1

4.7.2.1. Fun

A Friedman test comparing control schemes on fun rankings for all three games showed no significance. Rankings counts for most fun scheme can be found in Figure 19.



Figure 19. Scheme Rankings, Most Fun

4.7.2.2. Comfort

Significance was demonstrated when running a Friedman test on the comfort rankings for Donkey Kong ($\chi^2(2) = 14.625, p = 0.001$). We conducted post-hoc analysis using a Wilcoxon signed-rank test with Bonferroni correction applied ($p < 0.017$). These tests showed that binary ranked significantly higher than action ($Z = -2.911, p = 0.004$) and navigation ($Z = -2.982, p = 0.003$), but action and navigation were not significantly different in terms of comfort.

There was no significant difference in comfort between control schemes for Punch-Out.

There was a significant difference in comfort, depending on the control scheme, for Tetris ($\chi^2(2) = 12.933, p = 0.002$). Post hoc analysis determined that comfort ranked significantly lower when using navigation for Tetris compared to binary ($Z = -3.040, p = 0.002$) and action ($Z = -2.841, p = 0.005$). There was no significant difference in comfort between binary and action. Rankings counts for most comfortable scheme can be found in Figure 20.

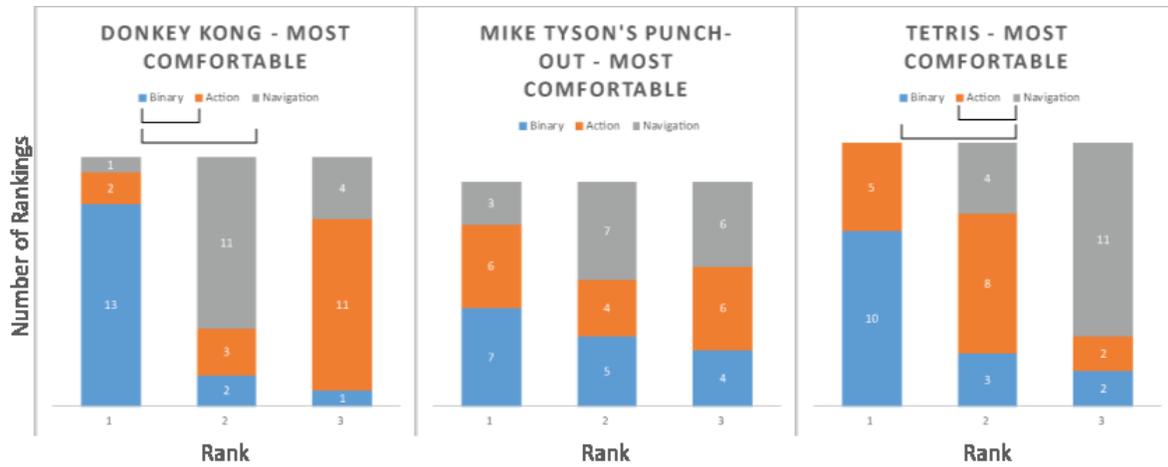


Figure 20. Scheme Rankings, Most Comfortable with Significant Differences

4.7.2.3. Frustration

Significance was demonstrated when running a Friedman test on the frustration rankings for Donkey Kong ($\chi^2(2) = 21.125, p < 0.0005$). We conducted post-hoc analysis using a Wilcoxon signed-rank test with Bonferroni correction applied ($p < 0.017$). Binary was significantly less frustrating than both action ($Z = -3.380, p = 0.001$) and navigation ($Z = -3.624, p < 0.0005$). There was no significant difference in frustration between action and navigation for Donkey Kong.

There was also a significant difference in frustration rankings between control schemes for Punch-Out ($\chi^2(2) = 10.500, p = 0.005$). Binary was significantly less frustrating than navigation ($Z = -3.051, p = 0.002$). All other comparisons showed no significance.

Finally, there was a significant difference in frustration between control schemes for Tetris ($\chi^2(2) = 24.400, p < 0.0005$). Binary was significantly less frustrating than both action and navigation, and action was significantly less frustrating than navigation ($p < 0.017$). Rankings counts for most frustrating scheme can be found in Figure 21.

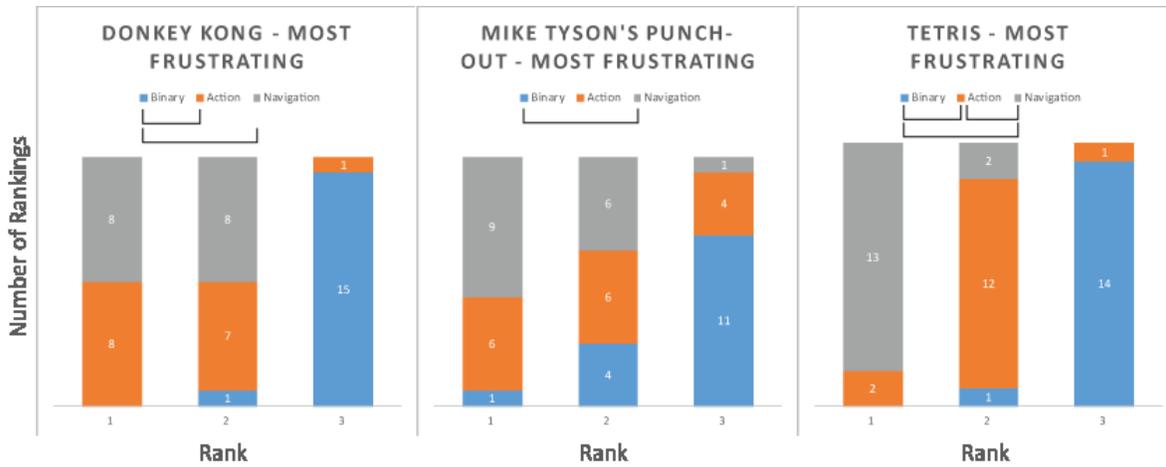


Figure 21. Scheme Rankings, Most Frustrating with Significant Differences

4.7.2.4. Naturalness

Significance was demonstrated when running a Friedman test on the naturalness rankings for Donkey Kong ($\chi^2(2) = 18.375, p < 0.0005$). We conducted post-hoc analysis using a Wilcoxon signed-rank test with Bonferroni correction applied ($p < 0.017$). Binary was significantly more natural than both action ($Z = -3.198, p = 0.001$) and navigation ($Z = -3.666, p < 0.0005$). There was no significance in naturalness ranking for Donkey Kong between action and navigation.

There was no significant difference in naturalness between control schemes for Punch-Out.

Finally, there was a significant difference in naturalness between control schemes for Tetris ($\chi^2(2) = 20.933, p < 0.0005$). Binary was ranked as significantly more natural than both action and navigation, and action was ranked as significantly more natural than navigation ($p < 0.017$). Rankings counts for most natural scheme can be found in Figure 22.

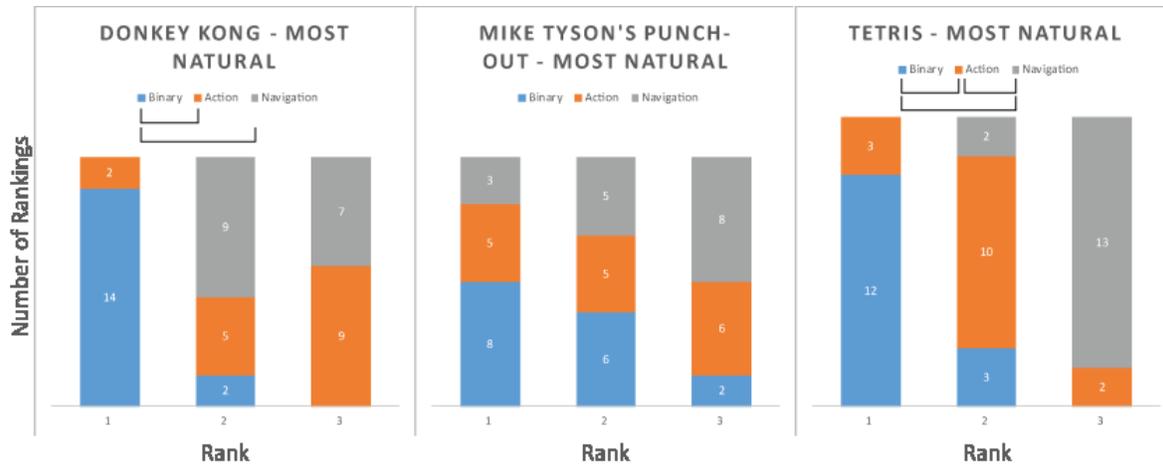


Figure 22. Scheme Rankings, Most Natural

4.7.3. User Experience Questionnaire

The User Experience Questionnaire (UEQ) asks participants to evaluate their experience by completing 26 ranking questions that compare various bipolar attributes. We separate these responses into 6 criteria: attractiveness, perspicuity, efficiency, dependability, stimulation, and novelty. The UEQ separates these 6 criteria into 3 high-level categories: attractiveness, pragmatic quality (perspicuity, efficiency, and dependability), and hedonic quality (stimulation and novelty). The scores are mapped from their original values (1–5) to values ranging from -2 to 2. This is done so that scores below 0 visually and numerically represent negative scores and scores above 0 represent positive scores as defined by the UEQ. We then categorized these scores and average them to determine the scores for each of the 3 high-level categories. We evaluated each of the schemes based on these three categories. A summary of the UEQ statistics can be found in Table 6.

| Measurement | Game | Binary Mean (SD) | Action Mean (SD) | Navigation Mean (SD) | Stats Binary/Action Binary/Navigation Action/Navigation |
|----------------------|-------------|---------------------|---------------------|-------------------------|--|
| Attractiveness | Donkey Kong | 1.05 (0.66) | 0.67 (0.91) | 0.37 (0.90) | not significant not significant not significant |
| | Punch-Out | 0.77 (0.49) | 0.66 (0.80) | 0.56 (1.08) | not significant not significant not significant |
| | Tetris | 1.06 (0.42) | 0.83 (1.09) | 0.11 (1.09) | not significant $p = 0.007$ $p = 0.021$ |
| Pragmatic Quality | Donkey Kong | 1.26 (0.27) | 0.57 (0.58) | 0.20 (0.70) | $p = 0.002$ $p < 0.0005$ not significant |
| | Punch-Out | 0.94 (0.50) | 0.51 (0.63) | 0.22 (0.71) | not significant $p = 0.009$ not significant |
| | Tetris | 1.43 (0.31) | 0.59 (0.64) | 0.12 (0.60) | $p = 0.001$ $p < 0.0005$ not significant |
| Hedonic Quality | Donkey Kong | -0.48 (0.53) | 0.59 (0.67) | 0.85 (0.39) | $p = 0.001$ $p < 0.0005$ not significant |
| | Punch-Out | -0.26 (0.58) | 0.97 (0.39) | 0.92 (0.53) | $p < 0.0005$ $p < 0.0005$ not significant |
| | Tetris | -0.44 (0.51) | 0.87 (0.64) | 0.70 (0.45) | $p < 0.0005$ $p < 0.0005$ not significant |

Table 6. Summary of UEQ Statistics for Experiment 1

4.7.3.1. Donkey Kong

A repeated measures ANOVA with a Greenhouse-Geisser correction determined that there was no significant difference between control schemes for attractiveness.

A repeated measures ANOVA with a Greenhouse-Geisser correction determined that the mean pragmatic score differed statistically significantly between control schemes ($F [1.830, 27.452] = 16.111$), $p < 0.0005$). Post hoc tests using the Bonferroni correction revealed that the binary scheme reached the highest mean perspicuity score, followed by the action scheme then the navigation scheme. The binary scheme scored significantly higher than both the action

scheme ($p = 0.002$) and the navigation scheme ($p < 0.0005$). The difference in score between the action and navigation schemes showed no significance.

Finally, we found significance between the mean hedonic quality scores ($F [1.975, 29.628] = 22.247$), $p < 0.0005$). Post hoc tests revealed that the navigation scheme scored the highest, followed by the action scheme, then the binary scheme. The mean differences between the binary scheme and the two other schemes were significant ($p < 0.05$). However, there was no significant difference between the action and navigation schemes. Donkey Kong's mean UEQ scores are displayed in Figure 23.

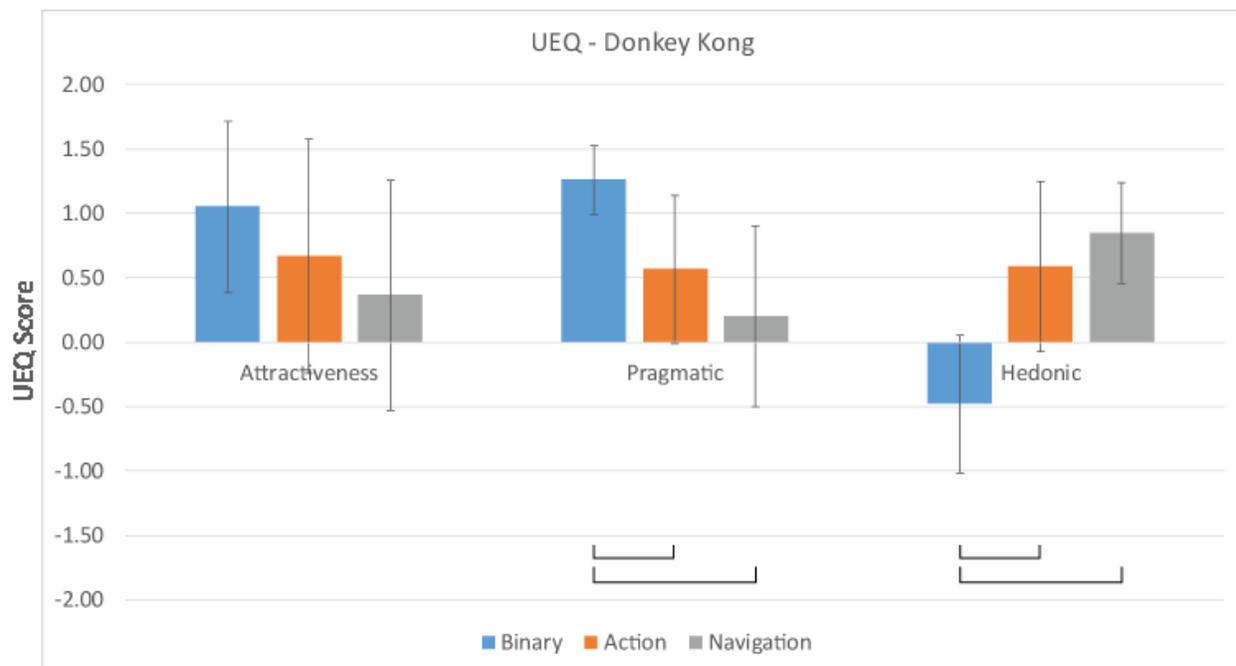


Figure 23. UEQ Mean Scores for Donkey Kong with Standard Deviation and Significant Differences

4.7.3.2. Mike Tyson's Punch-Out

A repeated measures ANOVA with a Greenhouse-Geisser correction determined that there was no significant difference between control schemes for attractiveness.

A repeated measures ANOVA with a Greenhouse-Geisser correction determined that the mean pragmatic score differed statistically significantly between control schemes ($F [1.922,$

28.824] = 6.487), $p = 0.005$). Post hoc tests using the Bonferroni correction revealed that the binary scheme reached the highest mean perspicuity score, followed by the action scheme then the navigation scheme. The binary scheme only scored significantly higher than the navigation scheme ($p = 0.009$). There was no significant difference between the binary and action schemes or the action and navigation schemes.

We found a significant difference between the mean hedonic quality scores ($F [1.561, 23.410] = 29.654$), $p < 0.0005$). Post hoc tests revealed that the action scheme scored the highest, followed by the navigation scheme, then the binary scheme. The mean differences between the binary scheme and the two other schemes were significant ($p < 0.05$). However, there was no significant difference between the action and navigation schemes. Mike Tyson’s Punch-Out’s mean UEQ scores are displayed in Figure 24.

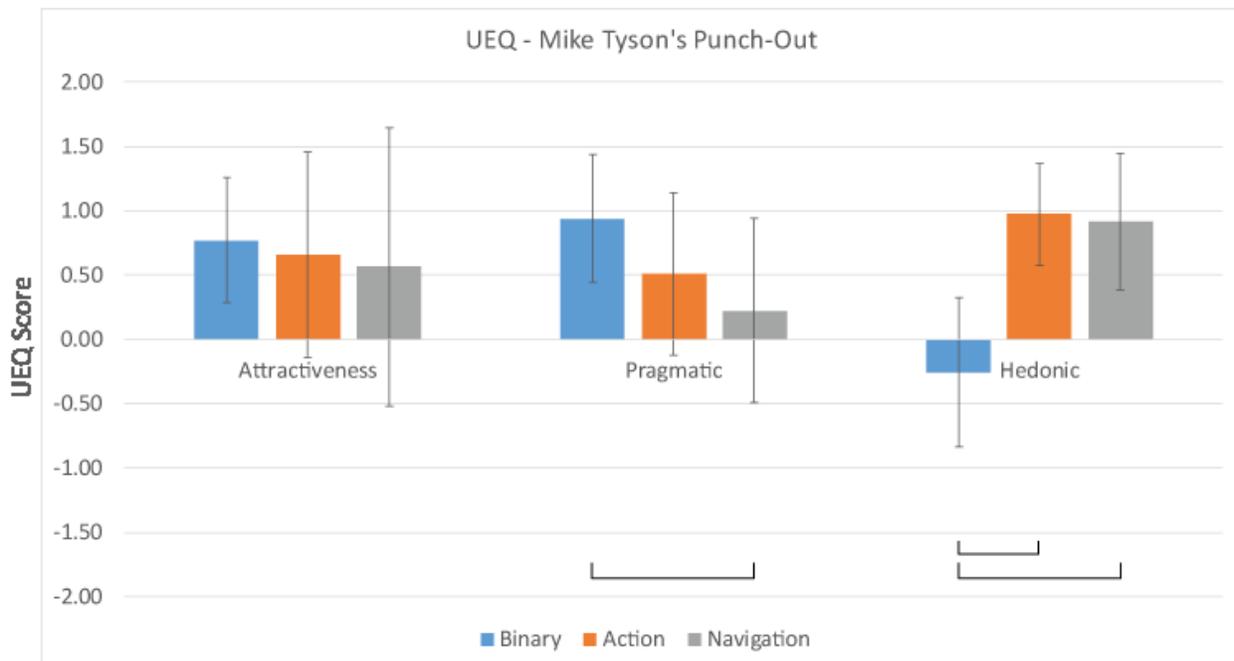


Figure 24. UEQ Mean Scores for Mike Tyson’s Punch-Out with Standard Deviation and Significant Differences

4.7.3.3. Tetris

A repeated measures ANOVA with a Greenhouse-Geisser correction determined that the mean attractiveness score differed statistically significantly between control schemes for Tetris ($F [1.954, 27.357] = 7.901$), $p = 0.002$). Post hoc tests revealed that the binary scheme scored significantly higher than navigation scheme ($p = 0.007$) and that the action scheme scored significantly higher than the navigation scheme ($p = 0.021$). We found no significant difference between the binary and action control schemes regarding attractiveness.

A repeated measures ANOVA with a Greenhouse-Geisser correction determined that the mean pragmatic score differed statistically significantly between control schemes ($F [1.999, 27.980] = 6.619$), $p < 0.0005$). Post hoc tests using the Bonferroni correction revealed that the binary scheme reached the highest mean perspicuity score, followed by the action scheme then the navigation scheme. The binary scheme score scored significantly higher than the action scheme ($p = 0.001$) and the navigation scheme ($p < 0.0005$). There was no significant difference in pragmatic quality between the action and navigation schemes.

We found a significant difference between the mean hedonic quality scores ($F [1.988, 27.827] = 30.639$), $p < 0.0005$). Post hoc tests revealed that the action scheme scored the highest, followed by the navigation scheme, then the binary scheme. The mean differences between the binary scheme and the two other schemes were significant ($p < 0.05$). However, there was no significant difference between the action and navigation schemes. Tetris's mean UEQ scores are displayed in Figure 25.

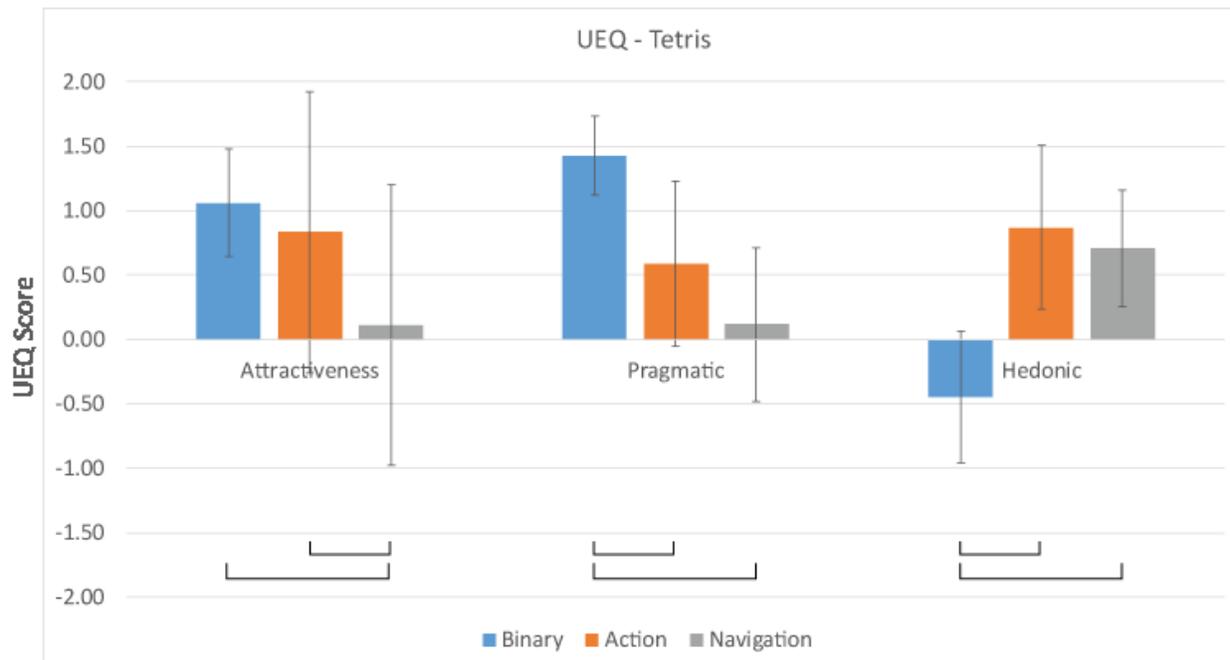


Figure 25. UEQ Mean Scores for Tetris with Standard Deviation and Significant Differences

4.7.4. NASA TLX

A repeated measures ANOVA with a Greenhouse-Geisser correction determined that the mean NASA TLX workload differed statistically significantly between control schemes for Donkey Kong ($F [1.870, 28.054] = 28.011$), $p < 0.0005$). Post hoc tests using the Bonferroni correction revealed that the workload for navigation was significantly higher than binary ($p < 0.0005$) and action ($p = 0.005$). It also revealed that the workload for action was significantly higher than the workload for binary ($p = 0.12$) in Donkey Kong.

Workload also differed statistically significantly between control schemes for Punch-Out ($F [1.762, 26.427] = 8.721$), $p = 0.002$). Post hoc tests show that the workload for binary was significantly lower than both action ($p = 0.001$) and navigation ($p = 0.012$). There was no significant difference in workload between action and navigation for Punch-Out.

Finally, workload differed statistically significantly between control schemes for Tetris ($F [1.960, 28.054] = 29.402$), $p < 0.0005$). Post hoc tests revealed that the workload for

navigation was significantly higher than both binary ($p < 0.0005$) and action ($p = 0.037$). There was no significant difference in workload between binary and action for Tetris. All mean workload scores for each game and control scheme are displayed in Figure 26.

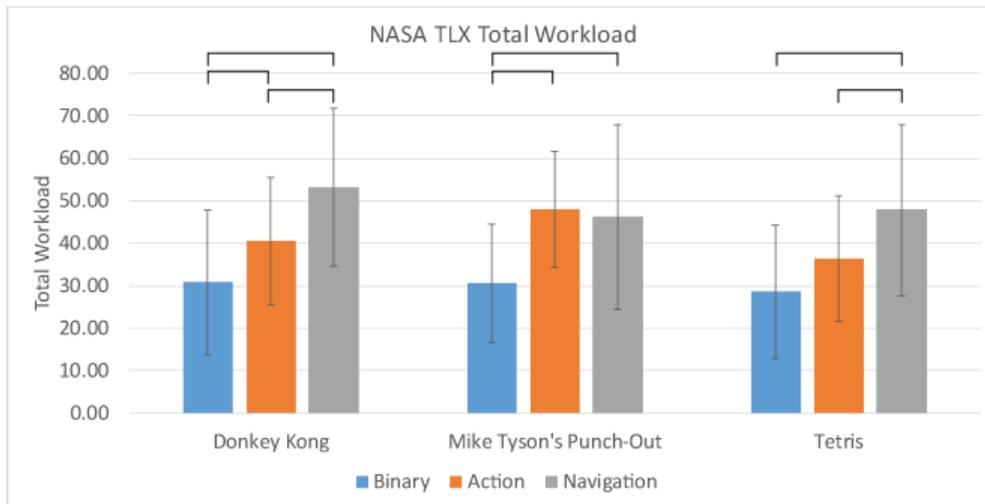


Figure 26. NASA TLX Total Workload and Significant Differences

4.7.5. Other

4.7.5.1. Post Questionnaire

We asked our participants three Likert questions at the end of the session to get a general sense of their feelings towards the controller. These answers ranged between 1 and 5, 1 being highly disagree and five being highly agree. We consider a participant in agreement if they responded with a 4 or a 5. 10 participants (63%) agreed with the first statement: “the controller was comfortable to use”. 16 participants (100%) agreed with the second statement: “the controller was fun to use”. Finally, 12 participants (75%) agreed with the last statement: “I would use this controller to play other games,” and only 1 participant disagreed (selected a 1 or 2).

We also asked participants which game they had the most fun playing. Mike Tyson’s Punch-Out was the most popular with 10 votes (63%), Donkey Kong was the runner-up with 5

votes (31%) and Tetris came last with only 1 vote (6%). Participants who chose Punch-Out justified their choice by claiming that the controls felt the most natural and they felt more immersed in the game when using bend/flex controls. Participants who chose Donkey Kong as the most fun game chose it because the controls were simple and “didn’t impede the gameplay” (Participant 5). Finally, the one participant who chose Tetris as their favourite game said it was “less complicated as it involved fewer actions” (Participant 1). Vote totals for most fun game can be seen in Figure 27.

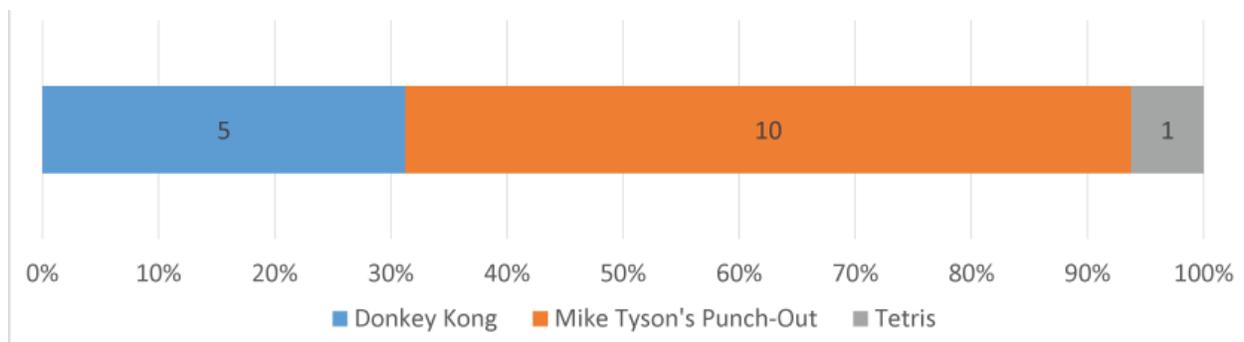


Figure 27. Vote Totals for Most Fun Game

We also asked users to describe their overall experience with the controller and which, if any, games they’d like to play with bend/flex control schemes. Participants were clear that the controller worked well for some games and not for others and also stated that it worked much better with simple actions as opposed to complex input such as navigation within the parameters of our experiment. Participant suggestions for potential game genres include: 3D flight games, racing games, 2D platformers (Mario, Sonic), arcade fighting games, rhythm games, and sports games. Three participants thought that the up/down and twist-left/twist-right were reversed at times. Five participants commented on the sensitivity and occasional unreliability of the input saying it was too sensitive thus resulting in some unpredictable input.

4.7.5.2. Input

We define input as any button pressed or any gesture performed. We recorded data every time a button or gesture began and ended. We only included input statistics based on input from trial 2 in order to remain consistent with the above results. Total input counts across all participants for each game-scheme combination can be seen in Table 7.

| | Binary | Action | Navigation |
|-------------|--------|--------|------------|
| Donkey Kong | 3400 | 3783 | 3609 |
| Punch-Out | 8720 | 8185 | 8896 |
| Tetris | 7175 | 4630 | 5822 |

Table 7. Total Input Counts Across All Participants

Mike Tyson’s Punch-Out tallied the most input of the three games. Donkey Kong required the least amount of input. Input totals between control schemes within Punch-Out and Donkey Kong remained somewhat constant. The input counts for Tetris were much more diverse. Input counts are displayed in Figure 28.

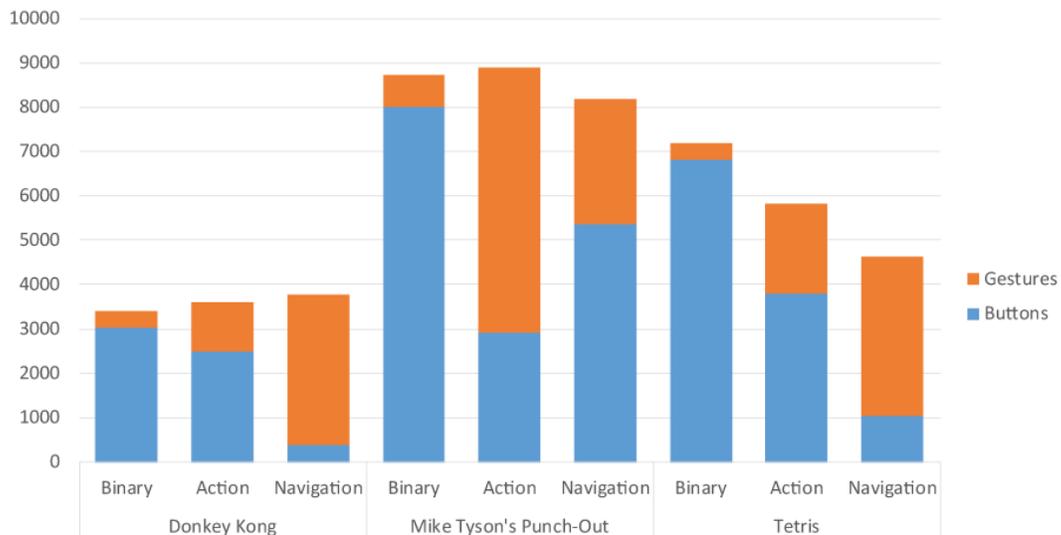


Figure 28. Total Input Count

We consider false-input as the participant pressing a button or completing a gesture that has no in-game mapping for the current scheme/game combination. Binary consistently had the falsest-input across games (DK = 10.7%, PO = 8.2%, and T = 5.1%). Navigation with Punch-Out had the highest percentage of false input (14.4%), however, unlike binary, this was not consistent over all three games. This means that users were using the directional pad buttons even though they had no functionality with that control scheme.

21.39% of gesture input for the action scheme in Punch-Out was false (i.e. participants attempted to bend up and bend down when they had no functionality). This false input may be due to the chaotic nature of this game-scheme combination. Participants were often very immersed in the game and were trying to punch as quickly and as frequently as possible causing a lot of unintentional input.

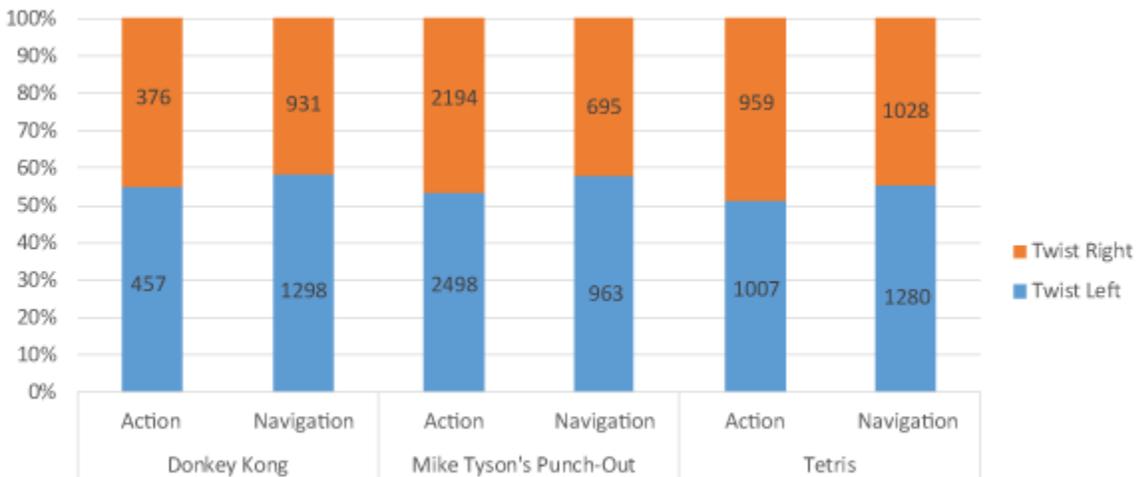


Figure 29. Twist Input Ratios

Participants used twist right and twist left, as seen in Figure 29, equally in situations where the user needed both directions (e.g. moving left and right in Tetris and Donkey Kong) and when direction was less important (e.g. rotating pieces in Tetris, punching in Punch-Out, and

jumping in Donkey Kong). Although rotating in different directions rotated the piece in different directions for the action scheme in Tetris and the rotation allowed participants to punch with different hands for the action scheme in Punch-Out, participants still could have succeeded using just one direction or the other.

4.8. Discussion

4.8.1. Binary Dominance

Binary consistently out-performed both action and navigation in Donkey Kong, Punch-Out, and Tetris, supporting H1. Binary ranked higher in comfort and naturalness, and lower in frustration, than the two gesture-based schemes across all three games. The UEQ displayed higher pragmatic scores for the binary scheme over the gesture-based schemes and consistently out-scored those schemes on attractiveness. These subjective test results support H2. The binary scheme's overall workload was lowest across all three games. Our results show that the binary control scheme, across all three games, performed better, ranked better, and required the least amount of work compared to the other two schemes.

We explain these results through participants' familiarity with standard game controllers and by the speed of completion of each type of input. First of all, we believe that participants' familiarity with standard controllers is the main reason why the binary scheme consistently out-performed and out-ranked our unique control schemes: all 16 participants were experienced playing video games using standard methods of input, while only two participants had limited experience using flexible input methods. Binary had the highest comfort rankings demonstrating that users were a lot more comfortable using the more standard method of input. Second, the time required to press a button (in place motion) is much lower than that of bending our

controller (3D movement). Hence, it took longer for participants to complete their tasks, producing a lower performance overall, which affected their game experience.

Based on these results, we anticipate that deformable gestures, specifically using our prototype, will not replace binary input for existing games, in part or in whole. However, we introduced the binary scheme in our experiment mainly to establish a baseline with current game controllers: our real objective with this experiment is to evaluate two novel control schemes using deformation as input.

4.8.2. Gestures Are Fun

Although the gesture schemes performed worse objectively, led to higher workloads, and that participants found them less comfortable and natural, they out-scored the binary scheme in the UEQ for hedonism. The hedonic category averages the stimulation and novelty scores of the UEQ. As this was an introductory study to our prototype, we did not design our tests to negate the novelty effect so this could have an impact on these hedonic scores. However, contrary to the action and navigation schemes, the binary scheme scored negatively in hedonism across all three games and we believe this would be case regardless of whether or not we tested with the novelty effect in mind. Participants evaluated action and navigation positively across all three games (the UEQ deems a score of 0.53 or greater as a positive evaluation (Laugwitz et al., 2008)). In addition, all three schemes across all three games received similar fun rankings from the game experience survey. This is interesting because the binary scheme significantly out-ranked gesture-based control schemes in all other categories, yet we found no significant difference for fun. While we hypothesized that action or navigation out-ranked binary in this category (H3), their close values demonstrate that using gestures does not decrease the amount of fun participants had while playing the games.

In addition, we believe that the frustration and discomfort the users experienced while using these schemes directly impacted the fun rankings for the gesture schemes. It is possible that by making the controller more comfortable to hold and use, and making the gestures more dependable in the back end, that the fun rankings will increase.

In the post-questionnaire, participants described their experience using the flexible prototype as riveting, innovative, and immersive. A participant stated that “using the bendable controller made the game more enjoyable compared to the regular button system”, and another participant added that the gestures “added another level (to the experience) which I found enjoyable”. Every participant agreed that the controller was universally fun to use with three quarters of them claiming that they would like to use bend and twist controls to play other games such as Mario Kart, Starfox, and Sonic.

4.8.3. Action Scheme Better Than Navigation Scheme

The action scheme, across most criteria for all three games, ranks higher than the navigation scheme as a method of gesture input with our prototype. Generally, performance was better using action but these differences were not proven to be significant which does not support H4. However, the workloads for Donkey Kong and Tetris using the action scheme were significantly lower, and the experience was better in many cases, which both support H5. The action scheme required less use of the gesture inputs than the navigation scheme across all three games (DK: 1 vs. 4, PO: 2 vs. 5, T: 2 vs. 3). We believe that introducing users to fewer new input methods led to more positive scores across all of our evaluations. Although mapping more in-game mechanics to gestures for the navigation scheme seems to be setting it up for failure, we were not sure, beforehand, that more mappings lead to poorer preference and performance. These schemes were also designed to represent a specific set of mechanics. Without fully mapping all

action and navigation tasks within the game to gestures, we would not be able to fully understand and compare their differences objectively or subjectively.

Simplicity of the games was a common theme within the participant's comments in the post-questionnaire. Participants mentioned that the controller "feel(s) best with simple games" and "the bending controls were very frustrating to use for navigation". The participant who made the last comment also claimed that the controller was fun when the gestures were used specifically for action. The simplicity of the action control scheme could be responsible for its lower workload scores, lower frustration rankings, and higher comfort rankings. We can attribute higher performance scores to having to learn less new input compared to the navigation scheme, as exhibited by the lower pragmatism scores for action.

Action schemes were unique between games, but results show that participants consistently preferred this scheme. In Donkey Kong, bending or twisting in any direction caused Mario to jump in game. Allowing users to move Mario around with the buttons allowed for more precision (which is necessary in Donkey Kong) and we believe that mapping jump to the gesture input simulates the urgency of jumping in the game. Often times, jumping is reactionary and users jump with little to no preparation in Donkey Kong which is why we believe gesture input fit so well: participants were able to trigger a jump by simply bending or twisting the controller using the quickest and easiest gesture for them. Bending or twisting the controller in any direction is easy and requires little to no thinking which is good as there is no time to think and it allows participants to focus more on the precise act of navigating the stage.

In Tetris, twisting the controller in either direction rotated the piece. Like in Donkey Kong, we believe that mapping the rotations to twists allows participants to focus on the more precision-based task of navigation using the more reliable binary input method.

4.8.4. Natural Mapping

Taking into account the gesture schemes only, we noticed that game/scheme combinations that ranked higher for naturalness (action and navigation in Punch-Out, action in Tetris) performed relatively better, had lower overall workloads, and were preferred by participants over the other gesture-based scheme.

Tetris was the only game where there was a significant difference in attractiveness between gesture control schemes, with action scoring higher. As mentioned above, action for Tetris also received one of the highest naturalness rankings. Twisting the controller in Tetris rotated the piece in the corresponding direction. We believe that this scheme ranked high in naturalness because the act of twisting is technically a rotation along the x-axis, hence naturally similar to rotating an option in the game. Multiple participants were able to predict how the action scheme would map before it was explained to them which illustrates its instinctive mapping.

Participant comments support their preference for action due to its naturalness. Participants who chose Punch-Out as the most fun, specifically commented on how natural twisting to punch felt. Participant 11 commented that twisting “gave a unique and tangible way to feel more immersed in the actual fight”, suggesting that naturalness of the scheme also increased immersion. However, both the action and navigation schemes for Punch-Out received similar scores across all of our evaluations. The act of leaning in either direction with our hands out in-front of us (similar to how boxers hold their hands out) is very similar to the input required to move left and right using the navigation scheme. The player could lean in either direction (in-game) using twist-left and twist-right while holding the controller in-front of them (which mimics the movement of the in-game boxer).

4.8.5. Scheme Consistency & Game Preference

Punch-Out behaved differently from the other two games: we did not find the action and navigation scheme's results to be significantly different in most cases. We did not find the binary scheme to be significantly different than the other schemes as often, when compared to Tetris and Donkey Kong. When combined with the fact that 63% of participants chose Punch-Out as the game they had the most fun playing using the flexible prototype, we find this game to be most successful for our novel controller. We believe that Punch-Out's consistency between control schemes is what led most participants to choose it as their preferred experience.

31% of participants selected Donkey Kong as their most fun game. We believe this to be mainly due to the inconsistencies between control schemes in Tetris with its navigation scheme scoring very low comparatively. Three participants specifically commented on the sensitivity and unreliability of the navigation scheme in Tetris. Another participant expressed their frustration when playing Tetris as they constantly felt the need to compensate their input by bending the controller in the opposite direction. Conversely, participants made no comments specifically criticizing the experience with our prototype and Donkey Kong. Consistency in controller reliability plays a key role in participant preference between games.

4.8.6. Task Types

Based on the results and comments made by participants, we should only be using discrete gestures for simple tasks and not for tasks requiring precision. Donkey Kong requires precision when navigating the stages as your character must be directly underneath the ladders to climb them. Users had difficulty doing so when using the navigation scheme. Moving pieces precisely around the screen in Tetris also proved difficult when users played with the navigation scheme mainly due to sensitivity issues. Both of these schemes scored much lower in pragmatic quality than their simple-gestured action scheme counterparts demonstrating a lack of

dependability. Workload for these two schemes were also significantly higher than their action scheme counterparts. A participant stated that “the bending controls were very frustrating to use for navigation, especially when dealing with precision”, referring to the precision required in both Donkey Kong and Tetris. Another participant summed up their experience by stating that “bends are better for simple actions” and a third participant chose Donkey Kong as the game they had the most fun with because “jumping with the bend felt quite natural, it was fun to use and didn’t impede the gameplay”. This evidence suggests that the navigation schemes were not reliable when the game required precision. It also suggests the preference of gestures for simple tasks (such as jumping in Donkey Kong or rotating pieces in Tetris).

4.8.7. Ergonomics & Handling

Almost every participant in our study handled the prototype differently throughout their session. They usually modified their grip to best fit the current control scheme. For example, if they did not need to use the Action 1 or Action 2 buttons, but needed to twist a lot, then they gripped the right panel on the top and bottom instead of the standard way of holding a controller (gripping the front and back). We noticed two participants using this grip method for the action scheme in Punch-Out where twisting was the more frequent input method as it was mapped to punching, the main action within the game. We believe that unique grips, such as the example above, are a product of applying gesture controls to games not designed with gesture controls in mind. In most cases, designers create games to use most, if not all, buttons on the controller so a grip that limits your ability to press two of the main buttons would not be an option.

We noticed that participants often made comments about how uncomfortable the controller was due to its rigid shell and harsh corners. We believe that this caused the decrease in comfort rankings for our gesture-based schemes as it required a tighter grip when using those

schemes. Bevelled edges and rounded corners could be a possible solution to the discomfort and may increase the comfort rankings for the gesture-based schemes across all three games.

Finally, we noticed some interesting behaviour in regards to how participants tried to use the gesture input. We observed a wide range of bend and twist intensities: some users were very gentle with the controllers whereas others seemed to think that harder bends and twists would result in harder punches or higher jumps. Our current prototype does not measure bend magnitude, but will be part of our future work. We noticed two participants had difficulty grasping the concept of twist and would often lean side to side or rotate the controller instead of twisting it. Finally, in Punch-Out, participants were trying to bend and twist the controller at the same time which resulted in some interesting deformations (even though we only registered one gesture input at a time). None of the schemes required bending and twisting at the same time.

4.9. Summary

We conducted an experiment to test the capabilities and seek uses for our flexible gaming controller. We chose to test it using three classic arcade games: (1) Donkey Kong, (2) Mike Tyson's Punch-Out, and (3) Tetris. In order to test the different uses for our prototype, we split up the mechanics of each game into two categories: (1) action and (2) navigation. We created one control scheme, for each game, that used gestures for the action mechanics and another that used gestures for the navigation mechanics. We also created a scheme that just used buttons, similar to a standard NES controller. We hypothesized that the binary scheme would perform the best and receive the highest pragmatic quality scores. We also hypothesized that the gesture-based schemes would be more attractive, fun, and novel. Finally, we hypothesized that the action scheme would perform better than and out-rank the navigation scheme. Our results revealed that the binary scheme did out-perform the gesture-based schemes and was the least frustration and

required the least amount of work to use. Results also showed that the gesture-based schemes were more stimulating and novel, but were not necessarily more fun or attractive. These results represent an introductory look at our hybrid control schemes and influence regarding the novelty effect was not taken into consideration however further research could help determine its effect. In conclusion, H3 was partially verified as gesture-based schemes did receive significantly higher hedonic quality scores but were no more attractive or fun than the binary scheme. The action scheme received better feedback than the navigation scheme overall, but feedback was more based on the task. Our participants preferred simpler tasks that used gestures. When the games demanded too much bending and twisting from participants, performance and preference went down. The most naturally mapped the gestures were, the more attractive and appealing they were to participants. Participants suggested using bends and twists for different types of games, like racing games, so we decided to test this in experiment 2.

Chapter 5. Experiment 2: Continuous Flexible Input Methods in 3D Racing Games

5.1. Overview

In experiment two, our goal was to use our prototype to test continuous gestures in games. Ahmaniemi et al. (2014) in their exploration of what bend gestures are good for came to the conclusion that continuous gestures work well to control parameters requiring either bipolar input or high resolution input. Bipolar input represents the input required to control two opposing events within the software. An example of this is pressing right on your keyboard to move an on-screen avatar to the right, whereas pressing left moves the same avatar the opposite direction, to the left. High resolution input represents any type of input that has multiple levels of magnitude. A trigger, for instance, can be held all the way down, half way down, a quarter of the way down, or not down at all, with an infinite number of other positions along the way. Unlike in experiment one, where we tested discrete gestures, we explored how continuous gestures could play a role in video game control mapping using bipolar input to control high resolution parameters.

Our secondary goal was to find games or game mechanics that can make better use of gestures. We decided to test a 3D racing game based on the comments made by participants in our first experiment. Our participants collectively suggested that a racing game could work well with gesture-based input given the natural similarities between steering and twisting our prototype. We believe that a racing game has potential to test continuous gestures and new, more advanced, game mechanics.

We are testing the same prototype as in our first study. We selected to test control schemes categorized the same way as in experiment one in order to see if results would stay consistent with the use of continuous gestures. We focus on a single game to investigate it in more depth. We also noticed that switching between games and schemes in the first experiment was sometimes overwhelming for participants and prevented them from mastering the schemes. Here, the mechanics of the game remain the same, so participants can focus on the control schemes.

5.2. Game

Participants played a 3D racing game called “O.R.B.S.” (Shorey, Nguyen, Dubord, Mukama, & Yang, 2014). In “O.R.B.S.”, players race artificially intelligent robots from point A to point B, making use of power-ups and weapons along the way. We have designed two custom tracks which we used to test two unique mechanics found in racing games: (1) speed, and (2) precision.

We designed the speed track, as seen in Figure 30, to test the participant’s ability to get from point A to point B in as little time as possible. The track is easy to navigate with few sharp turns. It has 27 power-up platforms scattered throughout. Each of these power-up platforms has a boost power-up available, which the participant can activate at any time. The boost power-ups can stack and become more powerful up to a maximum level of 3. Using the boost causes the racer to accelerate forward at a faster rate than normal for a pre-determined amount of time. We encourage participants to pick up and use as many of these boosts as possible in order to finish the race as fast as they can. The speed track is pink and blue with a daytime skybox in order to distinguish it better from the precision track.



Figure 30. In-Game View for the Speed Track

To test the participant's handling of the in-game racer, we designed the precision track, as seen in Figure 31. We still encouraged participants to get from point A to point B in as little time as possible, but were also told to get there with as few collisions with obstacles and barriers as possible. This track has a lot of sharp turns and obstacles intentionally placed to force participants to be patient and be more precise with the controller. There are no power-up platforms on this track as there are no areas within the track where boosting would be beneficial. The precision track is green and yellow with a sunset skybox in order to distinguish it better from the speed track. An aerial view of both tracks can be found in Figure 32.



Figure 31. In-Game View for the Precision Track

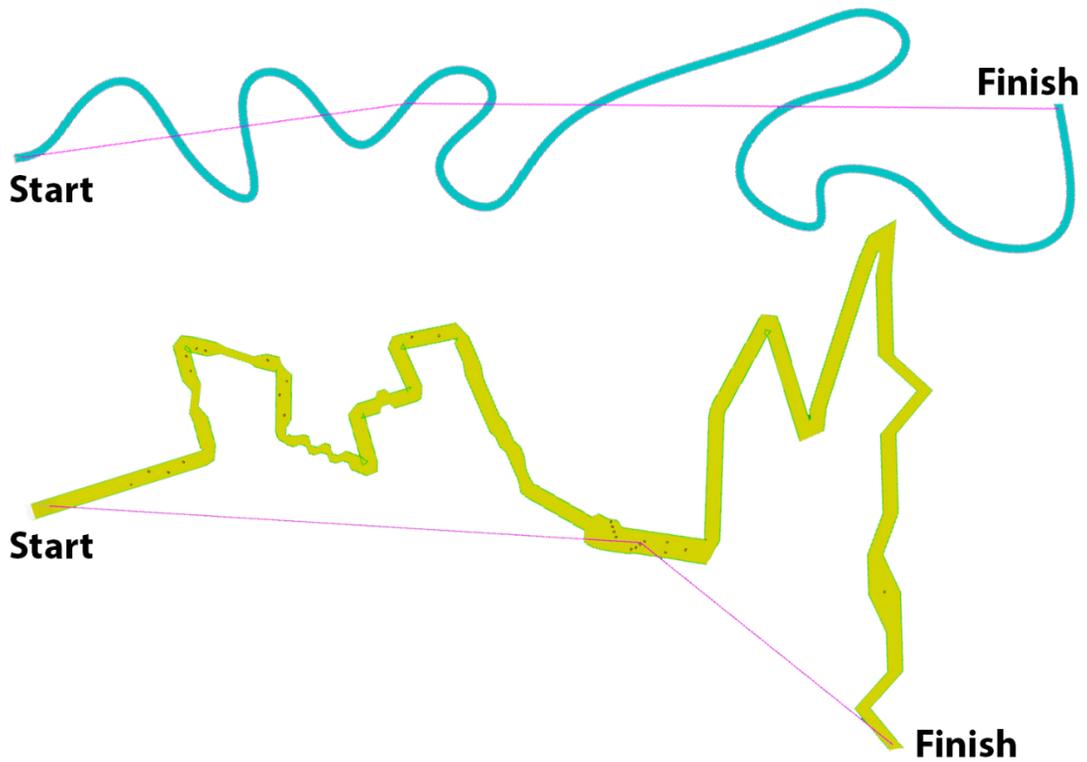


Figure 32. Aerial View of the Speed (Top) and Precision (Bottom) Tracks

We also created a practice arena for users to race around in before each trial in order to get used to each of the three control schemes. This arena is square, has no finish line, and is full of obstacles and power-up platforms so that the participant has a chance to practice all of the mechanics required in-game for each of the control schemes. We gave them this time to practice as we are not testing learnability and want them to be familiar with the controls before they attempt to get their best time and least number of collisions. They were allowed to practice as long as they wanted until they felt comfortable with how the scheme worked.

We are collecting event data from within the game code. There are 5 possible events with the following IDs: (0) input (1) collect boost (2) use boost (3) collision, and (4) end race. Every time an event takes place, we output a line to a custom data array. Each line contains the participant ID, the track, the scheme, the event ID, the key pressed (if relevant), a Boolean stating if that input was false input, the position of the racer at the time of the event, the speed of the racer at the time of the event, the boost level (if relevant), and finally the time stamp of the event. This array is then output into a .txt file with a unique name for further use in the analysis.

5.3. Control Schemes

Similar to our first experiment, we are using three unique control schemes that we named (1) binary (2) action, and (3) navigation (Figure 33). The binary control scheme uses only the digital input on the controller. Right and left on the directional pad control the direction of the racer. Up and down on the directional pad activate the current power-up. The left action button controls the racer's acceleration and the right action button slows the racer down. We designed this control scheme to be similar to typical control schemes found in older 3D racing games (before joysticks and triggers were built into game pads).

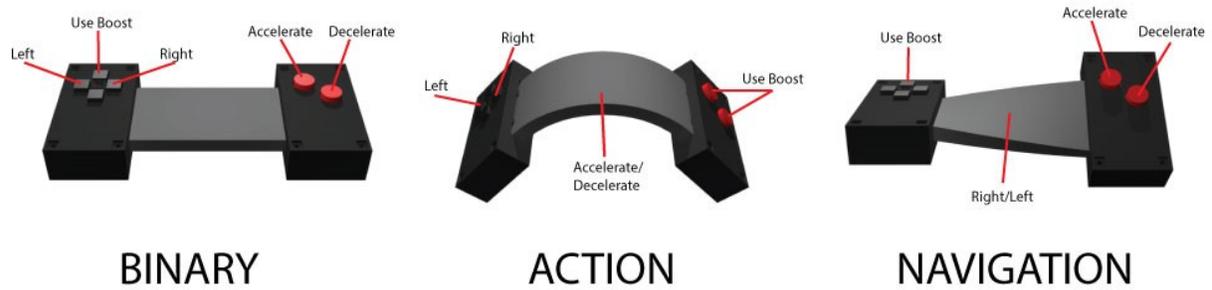


Figure 33. Control Schemes for Experiment 2

The action control scheme is the first of two control schemes that use the continuous gesture input. Bending the controller up controls the racer's acceleration: the greater the magnitude of the bend, the faster the racer will accelerate. Bending the controller down will slow down the racer: again, the greater the magnitude of the bend, the faster the racer will decelerate. Left and right on the directional pad are used to turn the racer left and right and we use the two action buttons to activate the current power-up.

The navigation scheme is the second scheme that uses continuous gesture input. Twisting the controller left and right turns the racer left and right. The more the user twists the controller, the more extreme the turn is. The action buttons, similar to the binary scheme, control the acceleration and deceleration. Up and down on the directional pad activate the current power-up. There is a small dead-zone where bending and twisting do not register as being bent in order to prevent sensor noise from impacting the participant's control for the action and navigation schemes.

The controller goes through a calibration process for both gesture-based control schemes. At the beginning of each race, the participant is asked to hold the controller comfortably still and a rest value for each of the sensors is recorded over a period of ten frames. We use these rest values in order to determine when a bend is taking place and the magnitude of that bend. Minimum and maximum bend values are also recorded for each of the bend sensors. We use

these values to determine if the sensors are bent to their extreme highs or lows. For the action scheme, we calculate the bend values used by the game based on the average of the how close both sensors are to their maximum or minimum values (positive values when the sensor reads above the rest value, negative values when the sensor reads below the rest value). We calculate the magnitude of a twist to the right as follows: $\left[\frac{b_2 - r_2}{r_2 - \min_2} \right] - \left[\frac{b_1 - r_1}{\max_1 - r_1} \right]$, where 1 and 2 refer to the bend sensor, and where b = bend, r = rest value, min = minimum sensor value, and max = maximum sensor value. We calculate the magnitude of a twist to the left as follows: $\left[\frac{b_2 - r_2}{\max_2 - r_2} \right] - \left[\frac{b_1 - r_1}{r_1 - \min_1} \right]$, where b = bend, r = rest value, min = minimum sensor value, and max = maximum sensor value. There is a threshold of 100 that the sensor has to pass in order for the software to register a bend. The twist values are also interpolated using the 1-euro filter (Casiez, Roussel, & Vogel, 2012) in order to filter out any unwanted noise or values sent through due to technical difficulties. These noise values include values of 0 or 1023 that were sometimes sent to due slightly loose wires.

5.4. Hypotheses

We based our hypotheses for this experiment on the results and comments from participants in experiment 1. Similar to experiment 1, we believe that the binary scheme will perform the best across both tracks (H1). We tested binary again in order to provide a basis for which to compare performance and different qualities from the UEQ. Although we hypothesize that binary will perform better objectively, we believe that it will score lower in hedonic quality compared to the gesture-based schemes following expectations developed from experiment 1.

We believe that the gesture-based control schemes (action and navigation) will score higher for hedonic quality, as well as fun, over the binary scheme in all cases (H2). Although

gesture-based schemes did not prove to be significantly more fun in experiment 1, we believe that making use of continuous gestures in a game genre that participants specifically suggested will be enough to distinguish gesture-based schemes as significantly more fun.

Our third hypothesis is that the navigation scheme will perform better and participants will prefer it over the action scheme (H3) based on the comments participants made in the previous experiment. Many participants stated that racing games would make good use of the twisting mechanic as twisting feels very similar to steering.

5.5. Methods

We asked participants to fill out a pre-questionnaire in order to obtain ethnographic data and data related to their experience with flexible input and racing games in general. We tested each of three control schemes with both the speed and precision tracks for a total of six trials. We counterbalanced the two tracks, then we randomized by scheme. Participants played both tracks back to back with each scheme before moving onto the next. We allowed participants to learn the control schemes first, by driving around in a practice arena before beginning the actual trials for that scheme. This gave participants an opportunity to ask questions and get a feel for how each of the control schemes were mapped out. When the participant indicated they were ready, we started on the track. They completed two trials on the track using the current scheme, then answered questionnaires relating to this combination. They then proceeded to complete the next track with the same scheme followed by the same questionnaires. They were not given more time in the practice arena between tracks. This was repeated for all three schemes.

The first questionnaire we asked them to fill out after each scheme/track trials asked four Likert-style questions regarding subjective performance, naturalness, fun, and frustration in regards to the control scheme itself. The second questionnaire was the user experience

questionnaire (UEQ) which we used in the first experiment and this questionnaire was to be answered in regards to their experience in that trial as a whole. We updated our questionnaire to include seven possible answers (instead of five in our first experiment).

We asked participants to fill out one final post-questionnaire once they had completed all six trials. We designed this questionnaire to figure out which track-scheme combinations participants prefer. It asked questions regarding which scheme was best for each track, which track participants preferred for each scheme, how easy it was to control the racer for each track-scheme combination, a favourite track-scheme combination as well as a least favourite combination.

We asked participants to think out-loud while playing and a research supervisor made observations throughout the process. The supervisor left the room when participants were completing the questionnaires in order to relieve any pressure over the participant. The entire session took approximately 60 minutes. All questionnaires can be found in the appendices (9.2.2. Experiment 2). This work was approved by Carleton University's Research Ethics Board.

5.6. Participants

We had participants sign a consent form before beginning the experiment. This consent form can be found in the appendices in chapter nine (9.1.2. Experiment 2). Our 19 participants (9 female) were all right-handed apart from 2 left-handed participants and 1 ambidextrous participant (mean age of 23.26). 10 participants claimed to play video games frequently, while 8 play them occasionally with only 1 claiming to never play video games. 9 participants have used a flexible method of input in the past and 5 of those participants participated in experiment 1. Although these 5 participants had used our prototype in the past, they did not all perform better than the new participants in this experiment. On a scale of one to five (one being no experience

and five very experienced), the average self-assigned experience score for racing games was 3.16 (SD = 0.83). On a scale of one to five (one being no skill and five very skilled), the average self-assigned skill score for racing games was 2.74 (SD = 1.10). We presented the participants with a \$10 gift card to Tim Hortons when their session was complete.

5.7. Results & Analysis

5.7.1. Performance

We measured time, collisions, and boosts used for every participant for each track-scheme trial. Every participant played both tracks with every control scheme apart from one participant who did not play with the navigation control scheme ($n = n - 1$) due to unforeseen time constraints. We analyze the values for the second trial, to avoid measuring the initial learnability of the level/scheme combinations. A summary of the performance statistics can be found in Table 8.

| Track | Measurement | Binary Mean (SD) | Action Mean (SD) | Navigation Mean (SD) | Stats Binary/Action Binary/Navigation Action/Navigation |
|------------------|---------------------|---------------------|---------------------|-------------------------|--|
| Speed | | | | | |
| | Time | 113.1 (9.1) | 111.1 (14.8) | 161.4 (57.2) | not significant $p = 0.005$ $p = 0.004$ |
| | Collisions | 6.74 (3.2) | 10 (5.3) | 33 (11) | not significant $p < 0.0005$ $p < 0.0005$ |
| | Boosts Picked Up | 13.5 (3.1) | 14 (2.5) | 8.3 (3.6) | not significant $p < 0.0005$ $p < 0.0005$ |
| | Boosts Used | 11 (3.9) | 11 (3.6) | 6.7 (3.3) | not significant $p = 0.001$ $p = 0.001$ |
| Precision | | | | | |
| | Time | 121.1 (30.4) | 136.9 (29.6) | 204.76 (54.2) | $p = 0.012$ $p < 0.0005$ $p < 0.0005$ |
| | Collisions | 9 (9) | 11 (9) | 63.0 (25) | not significant $p < 0.0005$ $p < 0.0005$ |

Table 8. Summary of Performance Statistics for Experiment 2

5.7.1.1. Speed

We performed a repeated measures ANOVA with a Greenhouse-Geisser correction on the time for the speed track, and found significant differences between control schemes ($F [1.127, 19.157] = 13.767$), $p = 0.001$). We used post hoc tests using the Bonferroni correction to investigate pairwise comparison. The navigation times were significantly higher than the action ($p = 0.004$) and binary times ($p = 0.005$). Mean times for the speed track can be seen in Figure 34.

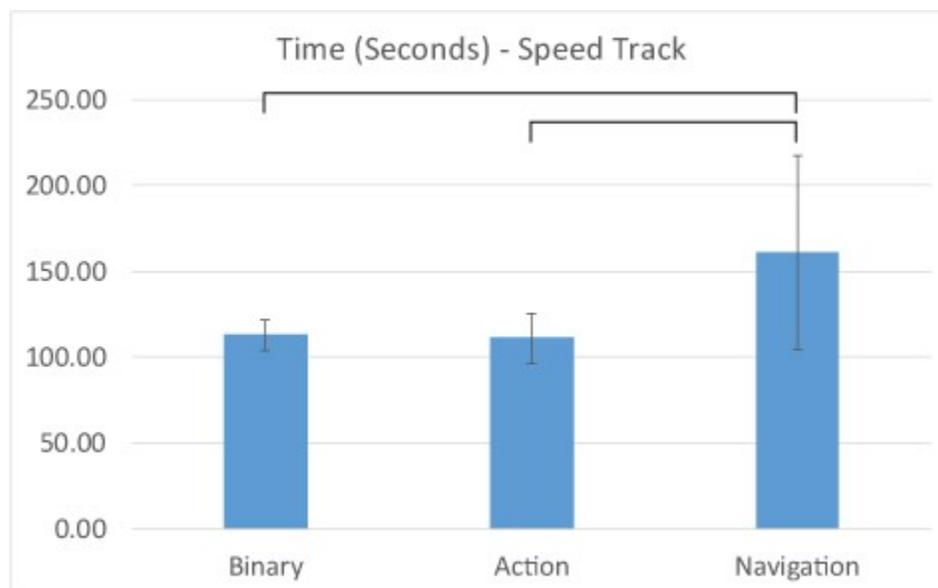


Figure 34. Mean Time for Speed Track with Standard Deviation

We performed a repeated measures ANOVA with a Greenhouse-Geisser correction on the number of boosts picked up for the speed track, and found significant differences between control schemes ($F [1.605, 27.286] = 21.856$), $p < 0.0005$). We used post hoc tests using the Bonferroni correction to investigate pairwise comparison. The number of boosts picked up while using the navigation scheme was significantly lower than the action ($p < 0.0005$) and binary ($p < 0.0005$). We also performed a repeated measures ANOVA with a Greenhouse-Geisser correction on the number of boosts used for the speed track, and found significant differences

between control schemes ($F [1.649, 28.029] = 15.865$), $p < 0.0005$). Again, using post hoc tests using the Bonferroni correction, we determine that the number of boosts used while using the navigation scheme was significantly lower than the action ($p = 0.001$) and binary ($p = 0.001$). Mean boost pickups and uses for the speed track can be seen in Figure 35.

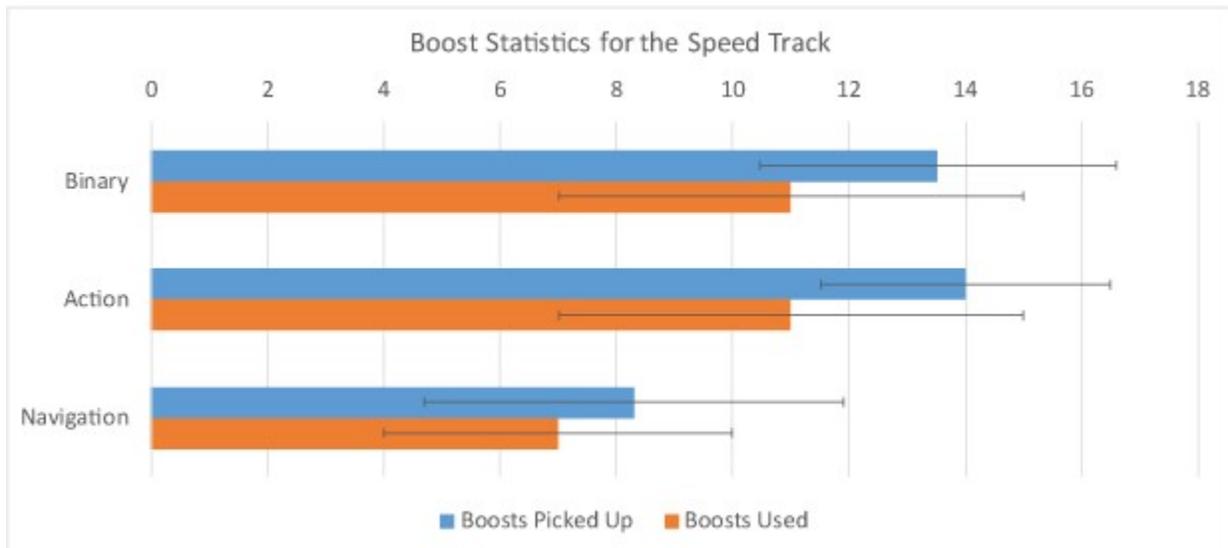


Figure 35. Mean Boost Usage for Speed Track with Standard Deviation

We performed a repeated measures ANOVA with a Greenhouse-Geisser correction on the number of collisions for the speed track, and found significant differences between control schemes ($F [1.319, 22.430] = 90.344$), $p < 0.0005$). We used post hoc tests using the Bonferroni correction to investigate pairwise comparison. Participants crashed significantly more when using the navigation scheme over the action ($p < 0.0005$) and binary schemes ($p < 0.0005$). Mean collision counts for the speed track can be seen in Figure 36.

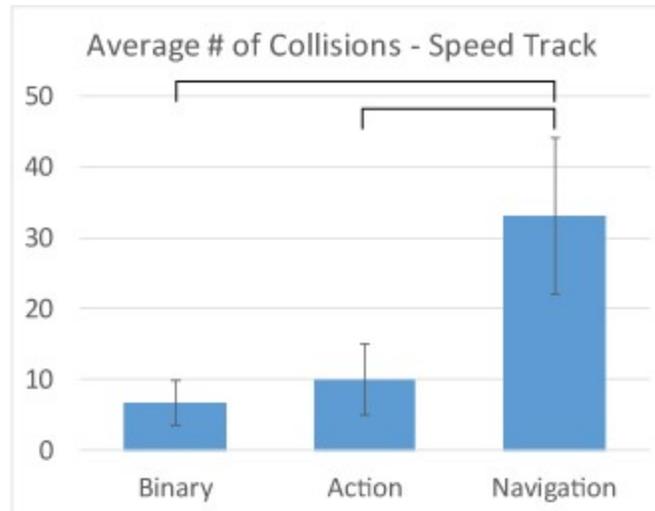


Figure 36. Mean Collision Count for Speed Track with Standard Deviation

5.7.1.2. Precision

We performed a repeated measures ANOVA with a Greenhouse-Geisser correction on the time for the precision track, and found significant differences between control schemes ($F [1.232, 20.939] = 29.849$), $p < 0.0005$). We used post hoc tests using the Bonferroni correction to investigate pairwise comparison. The navigation times were significantly higher than the action ($p < 0.0005$) and binary times ($p < 0.0005$). The action times were also significantly higher than the binary times ($p = 0.012$). Mean times for the precision track can be seen in Figure 37.

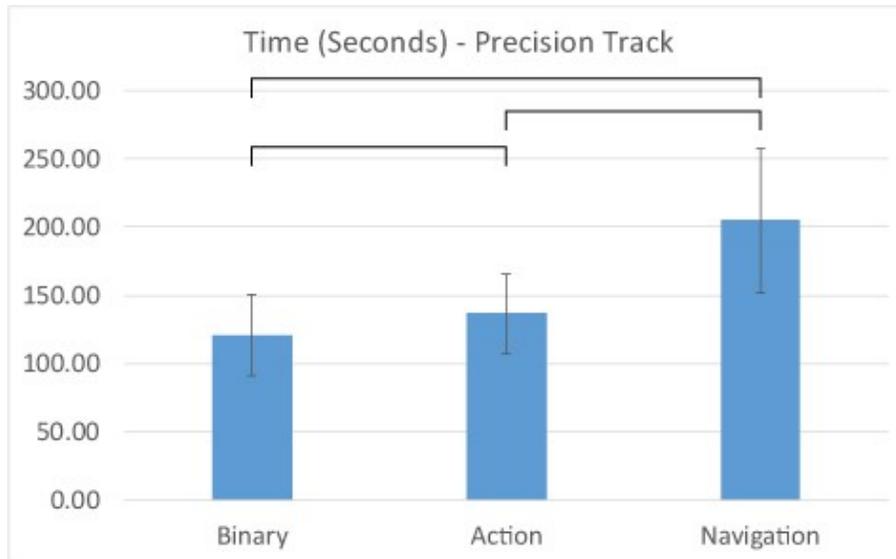


Figure 37. Mean Time for Precision Track with Standard Deviation

We performed a repeated measures ANOVA with a Greenhouse-Geisser correction on the number of collisions for the precision track, and found significant differences between control schemes ($F [1.050, 17.852] = 71.151$), $p < 0.0005$). We used post hoc tests using the Bonferroni correction to investigate pairwise comparison. Participants crashed significantly more when using the navigation scheme over the action ($p < 0.0005$) and binary schemes ($p < 0.0005$). Mean collision counts for the precision track can be seen in Figure 38.

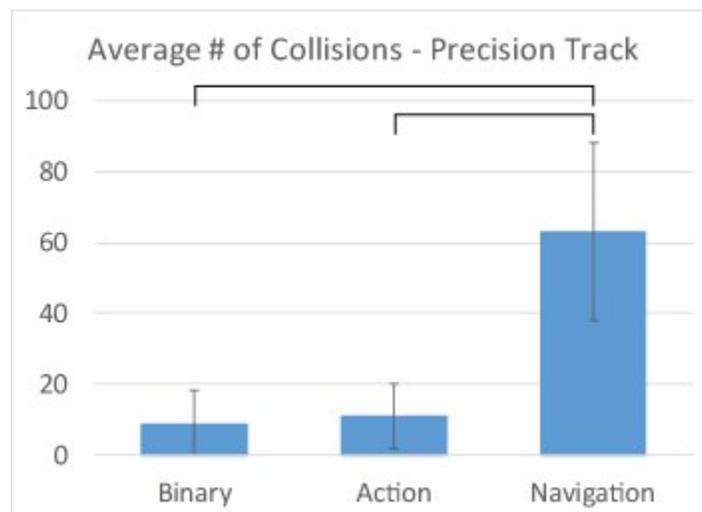


Figure 38. Mean Collision Count for Precision Track with Standard Deviation

5.7.2. Trial Experience

At the end of each trial, we asked participants to rate the track-scheme combination they just played in terms of how well they thought they performed, how natural they felt that combination was, how fun they thought it was, and how frustrating they thought it was. Each answer was a value between 1 and 5. We analyze the results of both tracks together. A summary of these statistics can be found in Table 9.

| Measurement | Binary Mean (SD) | Action Mean (SD) | Navigation Mean (SD) | Stats Binary/Action Binary/Navigation Action/Navigation |
|-----------------------|---------------------|---------------------|-------------------------|---|
| Perceived Performance | 4.26 (0.89) | 4.34 (0.91) | 2.68 (1.18) | not significant ($Z = -4.399, p < 0.0005$) ($Z = -4.635, p < 0.0005$) |
| Naturalness | 4.40 (0.97) | 4.05 (1.01) | 3 (1.27) | not significant ($Z = -4.354, p < 0.0005$) ($Z = -3.029, p = 0.002$) |
| Fun | 3.55 (1.25) | 4.16 (0.75) | 3.78 (1.13) | ($Z = -2.722, p = 0.006$) not significant not significant |
| Frustration | 1.92 (1.44) | 2.29 (1.33) | 3.16 (1.44) | not significant ($Z = -3.128, p = 0.002$) ($Z = -2.539, p = 0.011$) |

Table 9. Summary of Trial Experience Statistics for Experiment 2

Running a Friedman test demonstrated significance on the perceived performance rankings, ($\chi^2 (2) = 40.248, p < 0.0005$). We conducted post-hoc analysis using a Wilcoxon signed-rank test with Bonferroni correction applied ($p < 0.017$). These tests showed that navigation ranked significantly lower than binary ($Z = -4.399, p < 0.0005$) and action ($Z = -4.635, p < 0.0005$), but binary and action were not significantly different in terms of perceived performance. The perceived performance rankings can be found in Figure 39.

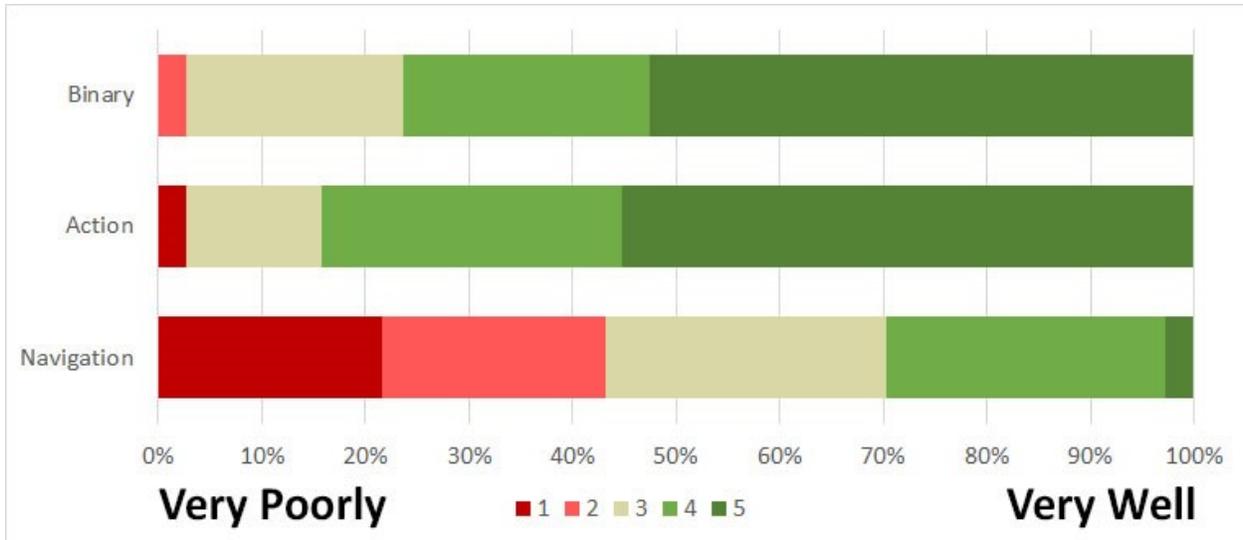


Figure 39. Track-Independent Perceived-Performance Rankings

We also found significance when running a Friedman test on the naturalness rankings, ($\chi^2(2) = 24.216, p < 0.0005$). We conducted post-hoc analysis using a Wilcoxon signed-rank test with Bonferroni correction applied ($p < 0.017$). These tests showed that navigation ranked significantly lower than binary ($Z = -4.354, p < 0.0005$) and action ($Z = -3.029, p = 0.002$). The naturalness rankings can be found in Figure 40.

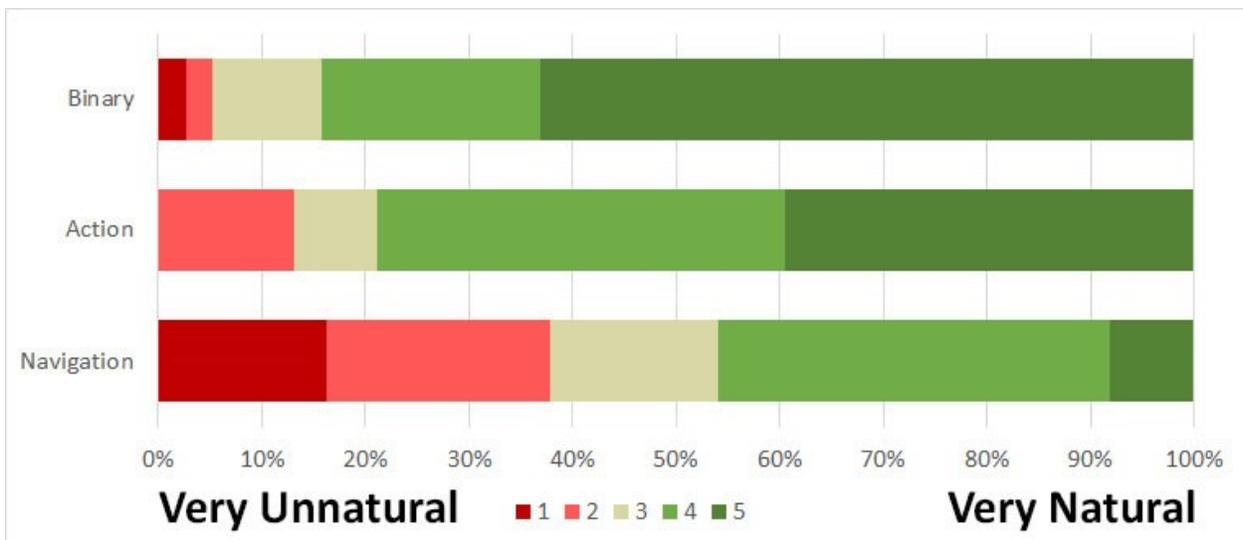


Figure 40. Track-Independent Naturalness Rankings

We found significance between schemes when running a Friedman test on the fun rankings, ($\chi^2(2) = 7.741, p = 0.024$). Post-hoc analysis reveals that the action scheme was ranked as significantly more fun than the binary scheme ($Z = -2.722, p = 0.006$). The fun rankings can be found in Figure 41.

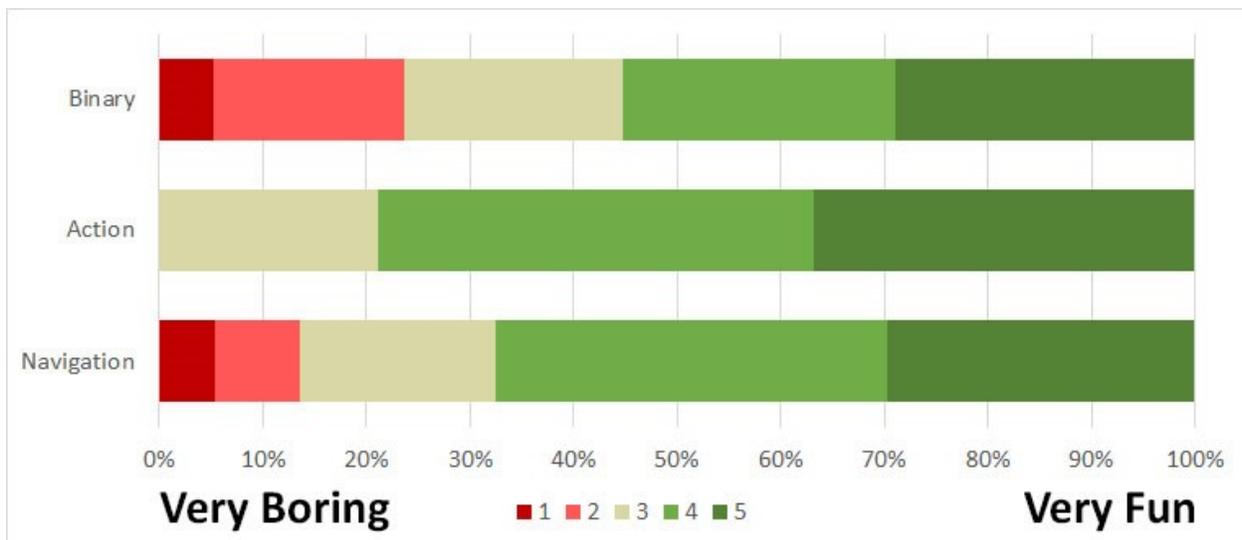


Figure 41. Track Independent Fun Rankings

Finally, we found significance between the frustration rankings for the control schemes, ($\chi^2(2) = 17.257, p < 0.0005$). We conducted post-hoc analysis using a Wilcoxon signed-rank test with Bonferroni correction applied ($p < 0.017$). These tests showed that navigation ranked significantly higher than binary ($Z = -3.128, p = 0.002$) and action ($Z = -2.539, p = 0.011$) for frustration. The frustration rankings can be found in Figure 42.

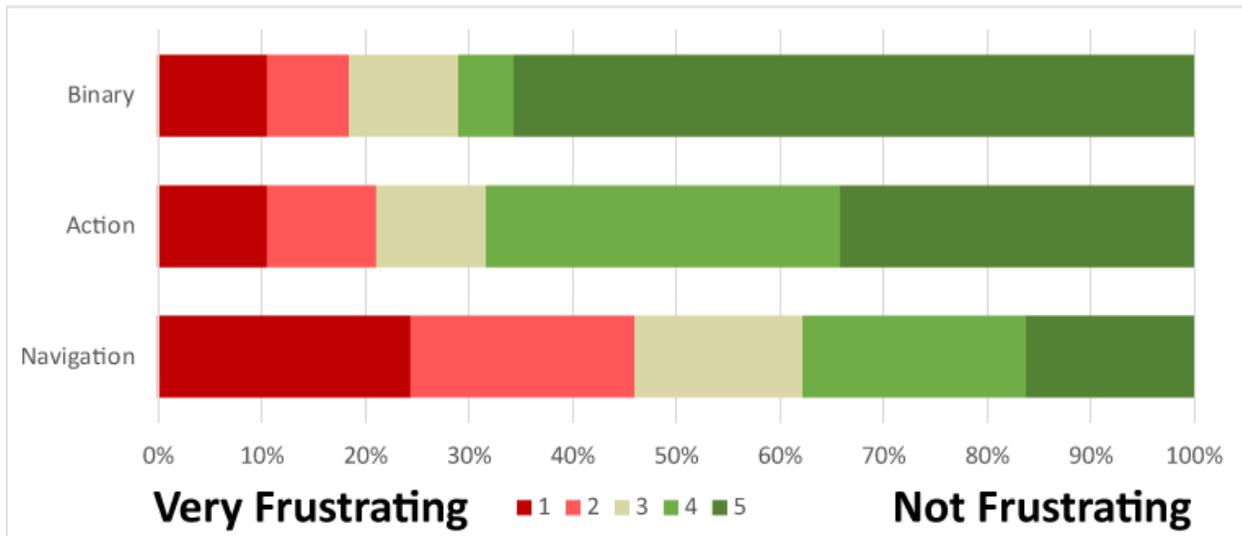


Figure 42. Track-Independent Frustration Rankings

5.7.3. User Experience Questionnaire

We performed statistical tests on the values of the speed and precision tracks put together in order to determine any overall differences in the UEQ results. We ran a repeated measures ANOVA with a Greenhouse-Geisser correction and determined that the mean attractiveness scores differed statistically significantly between control schemes ($F [1.933, 67.658] = 4.306$), $p = 0.018$). Post hoc tests revealed that the action scheme was overall significantly more attractive than the navigation scheme ($p = 0.027$).

The same test determined that the mean pragmatic quality scores differed statistically significantly between control schemes overall ($F [1.620, 56.685] = 23.143$), $p < 0.0005$). Post hoc tests revealed that the navigation scheme scored significantly lower than the binary scheme ($p < 0.0005$) and the action scheme ($p = 0.002$).

We also determined that the mean hedonic quality scores differed statistically significantly between control schemes overall ($F [1.832, 64.112] = 41.447$), $p < 0.0005$). Post hoc tests revealed that the binary scheme scored significantly lower than both the action ($p < 0.0005$)

and navigation ($p < 0.0005$) schemes. Overall UEQ scores can be found in Figure 43 and a summary of these statistics can be found in Table 10.

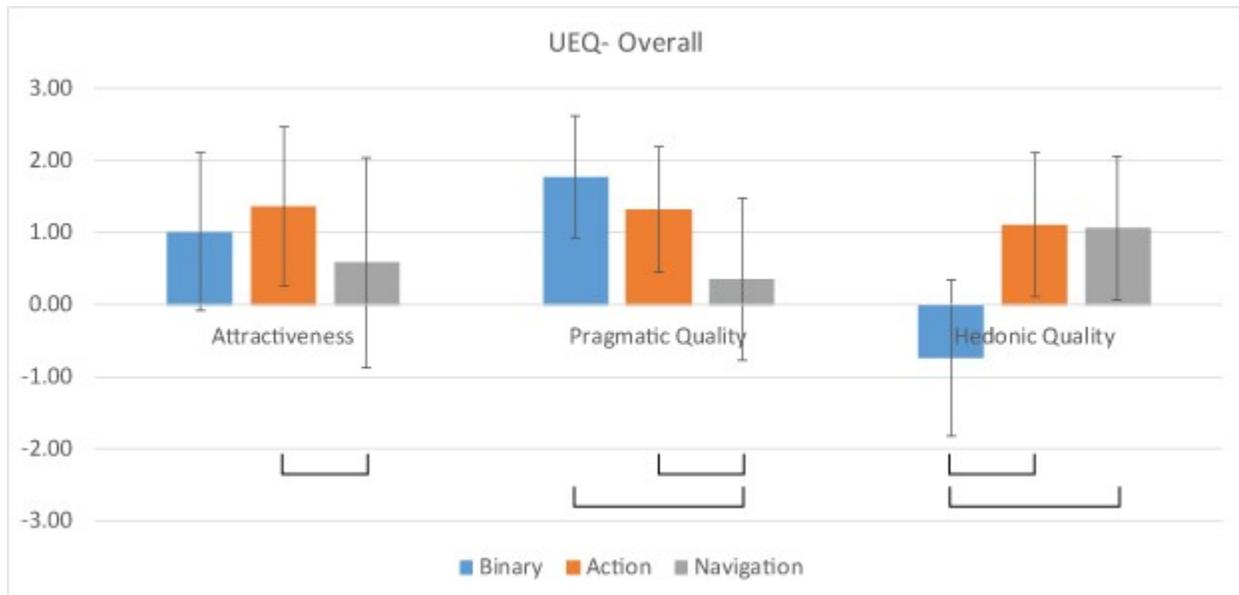


Figure 43. Overall UEQ Scores

| Measurement | Binary Mean (SD) | Action Mean (SD) | Navigation Mean (SD) | Stats |
|-------------------|------------------|------------------|----------------------|---|
| | | | | Binary/Action Binary/Navigation Action/Navigation |
| Attractiveness | 1.05 (1.07) | 1.45 (1.15) | 0.58 (1.46) | not significant not significant $p = 0.027$ |
| Pragmatic Quality | 1.81 (0.85) | 1.40 (0.93) | 0.36 (1.12) | not significant $p < 0.0005$ $p = 0.002$ |
| Hedonic Quality | -0.63 (1.18) | 1.18 (1.02) | 1.05 (1.00) | $p < 0.0005$ $p < 0.0005$ not significant |

Table 10. Summary of UEQ Statistics for Experiment 2

5.7.4. Post-Questionnaire

We asked users to rank the schemes based on preference for both tracks. A Friedman test comparing control schemes by rank for the speed track showed no significant preference.

Significance was demonstrated when running a Friedman test on the preference rankings for the precision track $\chi^2 (2) = 22.333, p < 0.0005$. We conducted post-hoc analysis using a Wilcoxon

signed-rank test with Bonferroni correction applied ($p < 0.017$). These tests showed that navigation ranked significantly lower than action ($Z = -3.672, p < 0.0005$) and binary ($Z = -3.586, p < 0.0005$), but action and binary were not significantly different in terms of preference. Scheme preference rankings for the speed track and precision track can be found in Figure 44 and Figure 45.

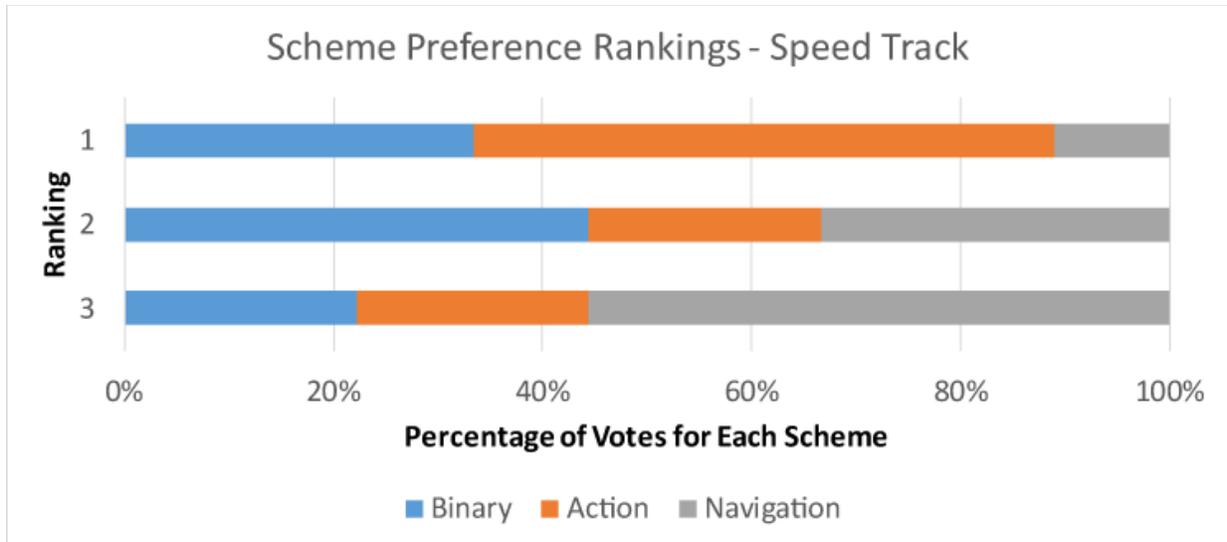


Figure 44. Scheme Preference Rankings Speed Track

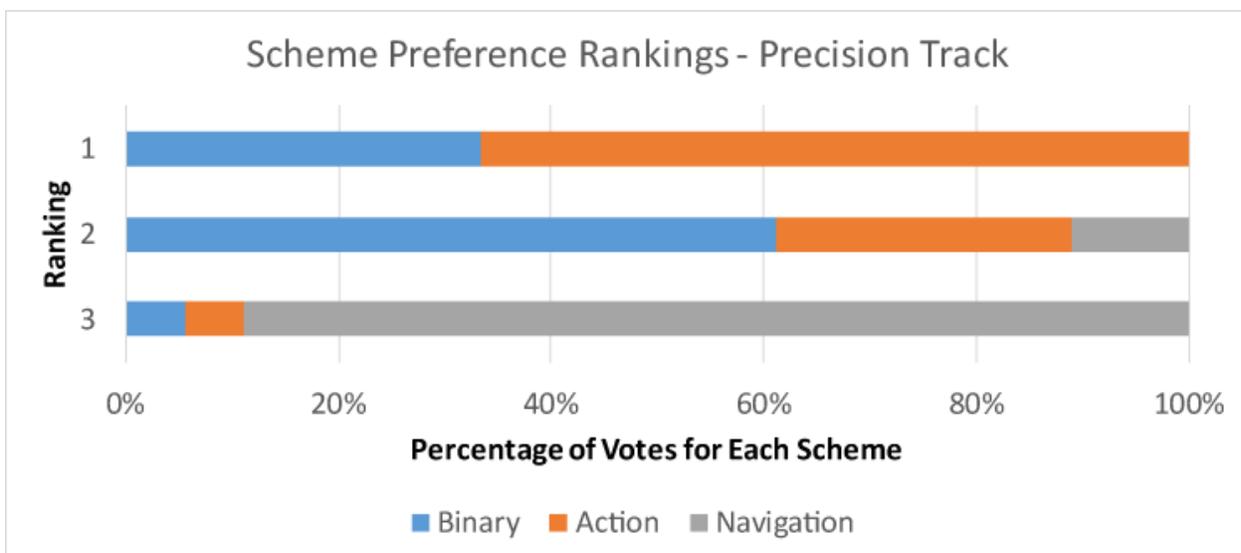


Figure 45. Scheme Preference Rankings Precision Track

We asked participants to choose which track they preferred for each scheme and compiled the results in Figure 46. Participants preferred the speed track, 33 votes to 24 votes (each participant had three votes as they were asked to choose their preferred track for each of the three schemes). Participants split their track preference for the binary scheme (9 votes to 10 votes), the action scheme weighed more towards the precision track (6 votes to 13 votes), and participants preferred the navigation scheme on the speed track (18 votes to 1 vote). The total number of votes received for track preference can be found in Figure 46.

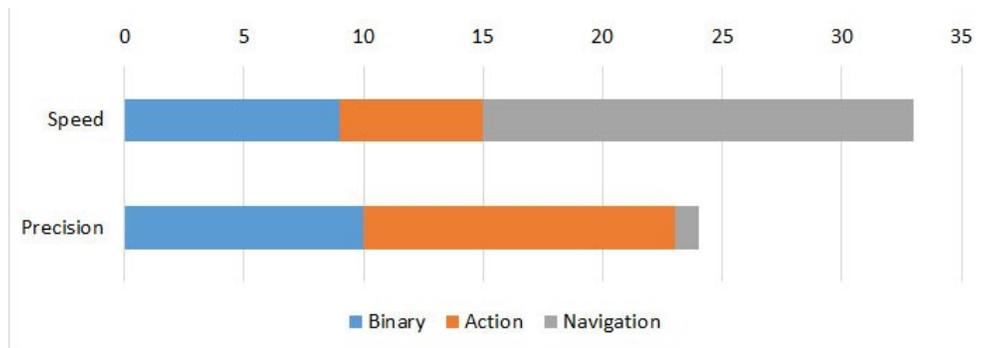


Figure 46. Track Preference by Scheme

Finally, we asked participants to select the track-scheme combinations they had the most and least fun using. Participants split the votes for most fun between the two tracks using the action scheme, each receiving 7 votes. The navigation scheme received 4 total votes and the binary scheme only received 1. The navigation scheme on the precision track received the most votes for least fun combination. Navigation received a total of 17 votes with both other schemes only receiving 1 vote each. Vote distributions can be seen in Figure 47 and Figure 48.



Figure 47. Total Votes for Most Fun Combination

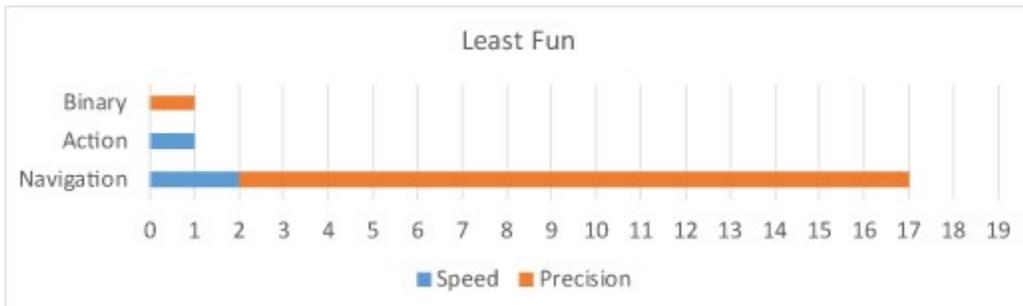


Figure 48. Total Votes for Least Fun Combination

5.8. Discussion

5.8.1. Binary is Still Familiar

The binary scheme is the most familiar and easiest control scheme to pick up and use successfully with a racing game, a result similar to that obtained in the first experiment. This confirms H1. Binary either performed the best or tied with the action scheme in all cases for both tracks. Given that all but one participant claimed to play games at least occasionally, the commonality of binary schemes within popular video games could be a cause of the higher performance scores this scheme received. We believe that participants' familiarity with binary input in other games made it easier for them to pick up and use more boosts on the speed track, lowering their completion times. The binary scheme received the highest pragmatic scores and, observationally, took the least amount of time to learn and become comfortable with on the

practice track. These results, alongside binary's low frustration and high naturalness scores, further support H1 in stating that binary is the easiest scheme to pick up and perform well with. Multiple participants stated in their comments that "binary was the most predictable" of the three control schemes.

Although participants could learn and perform well with the binary scheme, it was not the most preferred scheme to use. This shows a lack of correlation between performance and interest, fun, or attractiveness. Binary schemes are very familiar to gamers, and that lack of creativity within the scheme might be the cause of its low scores in terms of hedonic quality, attractiveness, and overall preference.

5.8.2. Gestures Are Still Intriguing

Our results reveal that the gesture-based schemes were more appealing and preferred over the binary scheme overall, confirming our second hypothesis. Action and navigation ranked highest in fun, hedonic quality, and received the most votes for the most fun scheme overall at the end of the study. Hedonic quality includes stimulation and novelty as descriptive factors and participants seem to have found the gesture-based schemes both stimulating and novel based on the results of the questionnaires along with their comments. Participants described gesture-based schemes as "unconventional, but what [they] were hoping for". They also stated that the gesture-based schemes "bring the user more into the actual gaming experience", commenting on their ability to immerse our participants into games as simple as our racing game O.R.B.S. We believe that these unique control schemes and input methods force users to focus more on what they are doing, possibly immersing them more in the entire experience. The freedom to bend and twist the controller in 3D space provides a natural interaction in terms of how people interact, almost instinctively, with everyday objects (with their hands, in 3D space).

5.8.3. Participants Still Prefer Action

We hypothesized that the navigation scheme would perform well, and participants would prefer it over the other schemes based on comments made by participants in experiment 1. This hypothesis was not supported by our experiment: the action scheme performed better and participants preferred it over the navigation scheme. This result is similar to that found by the performance of the navigation scheme in the first experiment.

Action consistently out-performed navigation on both tracks, received a higher perceived performance score, and was ranked as more natural and less frustrating than the navigation scheme. Action was also ranked as the most fun scheme overall. The UEQ results revealed that the action scheme was overall the most attractive scheme (even beating out binary) and scored higher in pragmatic quality than the navigation scheme proving that action was easier to pick up and play with. Action ranked as more preferred than navigation and participants selected it more often as the most-fun scheme (14 votes against 4 votes).

We found that, once again, action out-performed and out-ranked the navigation scheme in almost every area. In our experiment, we separated two core mechanics of racing games into the two gesture-based control schemes: (1) the action scheme allowed for continuous control of the speed of the racer, and (2) the navigation scheme allowed for continuous control of the horizontal direction (i.e. the steering) of the racer. These results demonstrate that users prefer using gestures to control the racer's speed and describe this scheme as "easy to learn," "more relaxing", and "adding excitement to the tracks". They liked being able to control their speed around corners and near boost pads allowing them to avoid collisions with walls and allowing them to pick up more boosts, which, in turn, lowered their completion time on the speed track.

The ergonomics of the scheme itself may explain the action scheme out-scoring navigation. A number of participants stated that “bending was more intuitive than twisting”. Participants had a hard time twisting the controller properly even after being shown how to do it. Participants would sometimes fold one of the rigid sides at a 45-degree angle in attempt to twist the controller but this would not register as a proper twist. Bending, however, is a simple gesture that people seem to be more familiar with, as there is only one way to do it (by folding both rigid sides either up or down). Another participant also stated that “action was simple enough to not accidentally move your fingers away from the required buttons”, commenting on the grip and handling of the controller itself. This statement implies that the movements required by the navigation scheme sometimes cause fingers to move away from buttons that participants used to accelerate and use boosts. Participants were able to reach all the buttons when using the action scheme.

5.8.4. Navigation is Difficult to Learn and Understand

The navigation scheme performed the worst and ranked the worst overall in experiment two. Although this scheme received some positive feedback in regards to hedonic quality, it did perform worst compared to the action scheme overall. While we did not focus on the learnability of the gestures in this experiment, our observations led us to believe that there is a steeper learning curve for gesture-based schemes, especially the navigation scheme. We noticed that participants took much longer in the practice course with the navigation scheme over the binary and action schemes. Participants also claimed that “the navigation scheme has a steep learning curve.” As a result of this learning curve, participants did not perform well with the navigation scheme after their limited amount of practice time. We did, however, notice large improvements in completion times and collision counts between their practice trials and recorded trials when

using the navigation scheme. This suggests that with practice, their performance could increase at a noticeable rate but this will have to be explored in a separate study.

We also noticed that participants struggled with learning how to twist the controller properly. Even after it was explained to them, some participants still would not twist it correctly, along a middle (invisible) axis. Poor twist input caused unexpected reactions in-game which could have led to both higher frustration rankings and lower fun and attractiveness scores. The few participants who did understand how to twist properly ended up ranking the navigation scheme higher and performed a lot better than those who did not.

Half of our participants stated that the navigation scheme was too sensitive whereas the other half seemed to think that the scheme was not sensitive enough. We believe that this division is the result of some participants not fully understanding how the navigation scheme worked. We told participants that in order to have more control over turning, with the navigation scheme, they should be accelerating. Twisting simply applies a force to either side of the racer so without any speed forward, the racer will only spin. We believe that this disconnect between the two mechanics is what made it frustrating for the participants. When participants did understand that accelerating helped with direction control, they did not think it was sensitive enough and expected more immediate results from twisting. The navigation control scheme uses a filter in order to smoothen the input values which causes a slight delay (<10 ms). Even though the delay was very small, it seems to have had a negative impact on how participants perceived this scheme.

We believe that the lack of understanding of the navigation scheme led to development of a negative connotation towards this scheme early on by participants. We observed almost all participants zig-zagging through the tracks as they overcompensated their turns in attempt to

straighten out their racers. This zig-zag pattern caused racers to collide more frequently and miss boost platforms which they managed to acquire successfully when using the binary and action schemes. This frustrating battle caused by the lack of understanding could have led to the very low UEQ scores the navigation scheme received and could also be the cause of participants ranking the navigation scheme as the least fun overall. We are interested in seeing how performance and preference could change if we give participants more time with the gesture-based control schemes, especially the navigation scheme.

5.8.5. Consistent Input as a Requirement

Users often have continuous control over their speed and turning in modern racing games such as Mario Kart 8 (Miyamoto, 2014) and Forza Motorsport (Turn 10 Studios, 2005). It is important to have continuous control over the racer's speed in order to slow down around corners and speed up when the track straightens out. It is also critical to have continuous control of the racer's direction in order to take turns at different angles and be able to precisely navigate through and around obstacles. We did not give participants continuous control over both speed and direction at the same time as our schemes split up speed and direction into two separate control schemes: (1) action and (2) navigation. Action had continuous control over the speed and discrete control of direction. Navigation had continuous control over direction and discrete control of speed. In both cases we had participants commenting on the lack of continuous control over either speed or direction. Performance and pragmatic qualities were also significantly lower for gesture-based schemes when compared to the binary scheme. This lead us to believe that consistent control over speed and direction simultaneously is a necessity in order to perform well in racing games. Based on our observations and comments made by participants, we believe that the lack of consistent controls had a negative impact on the frustration and fun rankings. This

problem could have been avoided by providing users with the ability to control both speed and direction continuously for the action and navigation schemes by implementing analogue sticks and triggers into the prototype.

One interesting observation is that the binary scheme out-performed both gesture-based schemes yet it did not provide the participants with any continuous input at all. We believe this is due to the participants' familiarity with standard control schemes and the similarity between how participants controller the speed and direction (they were both discrete instead of one being discrete and the other being continuous). This could suggest that control consistency is important in racing games regardless of whether or not it is discrete or continuous control.

5.9. Summary

We conducted a follow-up experiment to further test the capabilities and seek uses for our flexible gaming controller. Results from experiment 1 lead us to believe that our prototype had potential not only to be stimulating and novel, but to be attractive and fun as well. We chose to test a 3D racing game based on the feedback received in experiment 1. In order to differentiate it, we implemented continuous input and instead of testing multiple games, we tested multiple game mechanics within our game (speed and precision). We tested three control schemes on two different tracks, all presented in pseudo-random order to participants in order to eliminate learnability from influencing our results. We asked participants to fill out a pre-questionnaire, post-questionnaire, and two questionnaires after ever track/scheme trial. We hypothesized that the binary scheme would dominate in terms of performance, but participants would prefer the gesture-based schemes and they would be more appealing to participants. We also hypothesized that the navigation scheme would perform better than the action scheme based on comments made by participants in experiment 1. Results revealed that the binary scheme produced the

overall best performance results, but was not seen as a novel or stimulating experience. The gesture-based schemes were, for the most part, fun, novel, and stimulating with the action scheme scoring close to, if not better, than the binary scheme in terms of both performance and preference. The action scheme proved to work better than the navigation scheme and was the most attractive even scoring higher than the binary scheme. The navigation scheme was extremely frustrating and hard to control for participants and we believe the lack experience and understanding is the cause. We also believe that the results for the gesture-based schemes were lower than we had hoped due to the lack of continuous control over both the speed and direction of the racer. We would be interested in looking into the learning curves associated with the gesture-based schemes to see if, after some practice with them, results in performance and preference improve. We based this hypothesis on our observations of participants improving quite significantly between their practice trials for both tracks with both gesture-based schemes.

Chapter 6. General Discussion

In this chapter we step back and discuss our results overall, comparing our results from experiment one and experiment two. Based on the common results we suggest various design recommendations for researchers who wish to continue this type of research or for designers who wish to design games that implement deformation gestures. Finally, we discuss our limitations we encountered throughout our design process and experimentation.

6.1. Comparing the Results of our two Experiments

Overall, we discovered many consistencies in the results between experiment one and experiment two. Participants performed best with the binary scheme. In experiment one, the scores across the three games were higher, the death counts were lower, and the max levels were higher. In experiment two, the times were lower, the number of boosts picked up and used were higher, and the number of collisions were lower. We believe that it is fair to conclude that the binary scheme performs the best in these conditions, these conditions including the fact that participants had limited time to familiarize themselves with the control schemes and most participants had not used flexible input methods in the past. Button input is simple and this simplicity is important when mapping actions in video games. Users want to play games without thinking, they do not want to struggle with learning how the controls function. When implementing discrete gestures into games, designers should map these gestures to simple actions to not distract the user from the experience the game is trying to provide.

Secondly, gesture-based schemes were very intriguing to our participants. They were often excited to pick up and use the gesture-based schemes even if their performance was not as good with them. The action and navigation schemes, for the most part, received high hedonic quality scores across both experiments and participants often chose them as their favourite

schemes based on stimulation and fun. They did, however, score lower on pragmatic quality compared to the binary scheme which leads us to believe that more practice is necessary for participants to perform well with these novel schemes. When implementing deformation gestures into current games, find game mechanics that work similarly to how bends or twists work, and map those actions to the gestures. This represents a natural mapping which users enjoy based on our results. We also believe that deformation gestures should only be mapped to key-actions (actions that are critical in terms of game performance) if they represent a natural mapping. If no natural mapping is possible, we suggest to map them to novel in-game actions that add to the fun and enjoyment of the game, but are not critical in terms of performance such as honking your horn in a racing game or changing aesthetic properties within the game (colour, song, radio stations, etc.).

Next, the action scheme out-ranked and outperformed the navigation scheme in almost all cases between experiments. The navigation scheme, in experiment 1, often required more effort as more in-game actions were mapped to the bend gestures for the navigation scheme. These higher numbers could have resulted in a higher learning curve which then led to lower performance and, in some cases, lower preference ranks. All participants had the same amount of time to learn each of the schemes, so it is natural that if a scheme has more to learn than another, they would not have as much time to practise each of the mechanics with that scheme over a simpler one. For experiment two, the issue with the navigation seems to be the lack of understanding of twisting and how the twisting mapped to the game mechanics on screen. Again, the higher potential learning curve had an impact on how players performed and how they perceived the control scheme itself. On the other end, action was pretty simple for people to understand so people tended to perform better with that scheme and enjoy it more. We

recommend that when implementing twist continuously, it should be explained well to the user. The explanation should include how to physically twist the controller and how exactly it affects the game so that users are completely aware of how to use twist and what exactly it does. Twist is more forgiving when mapped discretely as it requires both sensors to be bent in opposite directions for a single frame and opposed to multiple frames when used continuously. When it is only required for a single frame, form is not as important and the action is performed immediately on screen confirming, to the user, that their twist was input correctly. Based on our results, we recommend mapping bends and twists to in-game actions, as opposed to in-game navigation. We also recommend mapping gestures to a minimal number of actions as more gesture mapping make the experience more complex leading to higher levels of frustration and worse performance.

Finally, the ergonomics of the controller and input methods caused a few issues in both experiments. Participants often had a rough time understanding how to perform a twist input, even after the experiment supervisor explained it to them. Their lack of understanding most definitely lowered their performance and increased their frustration in cases where we required twists. Some participants also thought that bending and twisting, combined with button use, was difficult because they would not hold our prototype like they would hold a standard controller since they needed a grip that allowed them to bend and twist more easily. This sometimes led to fingers not being on top of buttons when they needed to be. Although this was not a very common issue, it is something we should note for future design implementations. It is important to make sure that gestures do not take away from the core game experience by making it difficult to use the other methods of input.

In summary, both discrete and continuous gestures have potential in games, alongside buttons input, when mapped correctly. Results demonstrate that both discrete and continuous gestures are stimulating and novel and in some cases have potential to perform as well as standard methods of input (action scheme in experiment 2). We suggest that all gestures either be naturally mapped or mapped to simple in-game actions as these mappings resulted in better performance and a better user experience overall. If neither of those are possible, we suggest either not implementing gestures at all or implementing them in a non-intrusive way.

Implementing them in a non-intrusive way means mapping the gestures to in-game actions that are not required in order to complete the game or perform well: gestures are simply there to add fun by being mapped to mechanics that are supplementary to the necessary game mechanics.

Users should be taught how to perform gestures properly and should be shown how their gestures affect the game. It is important to remember that the addition of gestures should be helpful and fun, not confusing and frustrating and they should never take away from the core game experience. Introducing gestures into standard controllers allows for supplementary input options that would be accessible without needing to move your fingers off of the buttons. This leads to users being able to control more in-game mechanics without needing to move their fingers away from their natural positions. All of these recommendations apply to the development of new games and to the creation of control schemes for existing games.

6.2. Limitations

We came across a few limitations while prototyping and conducting our experiments. Our primary limitation was the unreliability of the bend sensors, and in some cases, the buttons. We encountered a lot of problems with sensor placement and security while prototyping. The nature of bend sensors is that they will be bent, so we needed to make sure that they could

withstand a lot of bending and twisting without moving too much, coming loose, or getting damaged. However, even with the sensors secure, the sensors tend to deform slightly with use, changing their base, minimum and maximum values (relative to the controller and how far users can bend them in either direction). We implemented a calibration system to automatically modify the base value for the sensors after every trial (or on demand by pressing a button). However, this did not change the maximum or minimum values which we needed in some cases to determine how close the sensor was bent in one direction or another from the base reading. The minimum and maximum values often changed by a factor of 1 to 100+ depending on how the prototype was left after the previous study. These changes in minimum and maximum values made the twists and bends either more or less sensitive depending on which direction the minimum and maximum values changed. We imagine a dynamic calibration system would work well with bend sensors but we did not manage to implement such a system. Often times, in experiment one, one or both bend sensors would be reading as either 0 or 1023 (the minimum or maximum) which caused unexpected reactions in-game. This was fixed in experiment 2 when we implement the 1-euro filter which filtered out extreme value changed between frames. Also, as mentioned, sometimes buttons would malfunction and require the users to press them down really hard in order for the Arduino to acknowledge the press. The hard presses then caused the button caps to become loose and eventually fall off. We would recommend a more professionally built prototype for future testing.

The second limitation we faced was the lack of an ability to track multiple gesture inputs at the same time. Participants could not bend and twist the controller at the same time, it was either bent or twisted. This limited what we could map the gesture-input too. Even though multiple gesture input was not implemented, we often saw participants attempting to bend and

twist at the same time in order to perform more than one action in game. It is possible for users to perform simultaneous actions in some of the games. We designed our control mappings to not require simultaneous gesture inputs, but some users were attempting bend and twist simultaneously either way as they were aware that various in-game actions could be performed at the same time. We believe that a system could be implemented that measures more than four gesture-inputs, but telling users to bend and twist at the same time is not intuitive and would not work using the software we implemented. Also, we believe that bending and twisting at the same time is not a clear way of describing input methods and instead believe that a new system would require different ways of describing the input methods which would include different degrees of twisting or bending. We also believe that in order to get more precise bend/twist information, we would require more than two bend sensors (although two sensors worked for us as we only tested four different gesture inputs).

Finally, we sought to determine where deformation gestures fit into gaming. We tested various genres of games using discrete and continuous gestures, but we left many genres untouched. We were not able to test every game genre as there are many. We were also not able to test complex game mechanics with our prototype as the prototype itself was quite simple. We believe that with a more complex prototype that implemented input methods such as analogue sticks, left and right triggers, or the ability to sense different degrees of the bend/twist, we could have tested more complex mechanics. This is an area of research that can be explored in the future.

Chapter 7. Conclusion

7.1. Overview

In this thesis, our goal was to determine if and where deformation input could fit in with standard gaming input methods. We wanted to test both discrete and continuous gestures alongside standard button input on a standalone prototype to see if users could make use of both methods simultaneously while still performing well and having a good time. We also wanted to see what types of games worked best with deformation input and which types of tasks mapped best to both bend and twist input methods. We created a flexible prototype with six buttons and four possible gesture inputs: (1) bend up, (2) bend down, (3) twist left, and (4) twist right. In our first experiment, we tested discrete gestures in three classic arcade games. We separated the in-game mechanics in terms of actions and navigation and assigned one control scheme to each of these (where, for the action scheme, gestures would control in-game actions and, for the navigation scheme, gestures would control in-game navigation). We also created a traditional control scheme that just made use of buttons that we named the binary control scheme. Overall, we tested three control schemes. We recorded score, max level, and deaths as participants played all three games with all three control schemes. We also measured subjective data regarding the scheme through questionnaires presented to participants throughout the testing process. We found that the binary scheme performed the best and required the least amount of work, but the gesture-based schemes were stimulating and novel. The action scheme performed better than the navigation scheme, was preferred, and required less work. In experiment two we tested continuous gestures using the same controller and same control scheme classification. We tested two tracks in a racing game: we designed the first track to test speed and the second track to test precision. Participants played both tracks with all three schemes. Again, we collected both

objective and subjective results throughout the testing process. Results revealed that once again, the binary scheme performed the best and gestures schemes remained novel and stimulating. Action performed significantly better than navigation, and was the most attractive and fun scheme. Participants had a lot of issues with the navigation scheme and understanding the ergonomics of twisting compared to bending.

discretediscretediscreteBy combining gesture input with standard input in our prototype, we created a user experience that was not only novel, but was stimulating and full of potential. Our efforts to discover which actions are best mapped to gestures and which are best mapped to buttons revealed some interesting results. Simple and naturally mapped actions, mapped to gestures, tend to be preferred amongst users and also perform significantly better than more complex and abstract actions. By allowing users to perform specific actions in-game using bends and twists, we allowed them to focus on more important in-game mechanics which led to better results. We received positive feedback from both discrete and continuous gestures in various cases. The actions associated with these cases are representative of the types of actions that should be mapped to gestures for the best results whereas game mechanics that performed poorly with gesture should be mapped to standard button input. Results show that gestures that are mapped well can increase immersion and stimulation. We believe that with sufficient practice, bend gestures will also have the potential to increase performance, in both old and new games alike, but further testing will need to be conducted. The combination of deformation gestures with standard button input gives users access to more methods of input without requiring them to move their fingers around to reach different buttons but the ergonomics of our controller need to be improved so that it can be bent easier using a more standard grip (as many participants held and bent the controller in very unique ways). Although technology such as the Kinect and Leap

Motion are capable of detecting similar gestures, we believe that more advanced and precise functionality can come from combining physical deformation gestures with buttons. We believe that our suggested design guidelines will aid researchers and game developers alike to improve on this hybrid technology in order to create game experiences where gestures are both preferred and perform well.

7.2. Future Work

A first avenue for future works relates to the learnability of bend gestures in combination with standard methods of input such as buttons. Our observations from experiment 1 and experiment 2 hinted at the presence of steep learning curves. We could test this by using the current prototype and getting participants to play the same game with the same scheme over a longer period, within one session or over multiple sessions.

It would be interesting to map continuous input to different mechanics in different genres of games. There are also two ways of implementing continuous input: (1) position control mapping and (2) rate control mapping (Strohmeier et al., 2016). We would use position control mapping in situations where the game would directly use the degree of the bend/twist. For example, we could use twisting to absolutely rotate a camera on the y-axis where the value of the twist is exactly the y-rotation value of the camera itself (i.e. twist to the max in one direction and the camera rotates as much as it can in the direction). This same situation, using rate control mapping, would actually spin the camera at different speeds and directions depending on how much and which direction the user was twisting the controller (i.e. twist the controller slightly to the right, and the camera would rotate to the right as a certain speed).

In our research, we used games that already existed or slightly modified versions of them in order to determine if gestures have a place in gaming when combined with standard input.

Although our results show that there is potential when it comes to fun, intrigue, and novelty, we believe that games designed specifically for bending and twisting could yield much different results. When testing games built for bending and twisting, we no longer need to focus on comparing gesture-based schemes to standard control schemes as the standard control schemes would no longer apply.

Finally, as mentioned earlier, we believe that a more reliable and sturdy controller, possibly with more degrees of freedom (in the sense of having more than four gesture inputs) would be something of value to create. We could then disregard any unreliable sensors and buttons and focus on the mapping of the sensors themselves. We could also test more complex games if we were to measure more than four gestures using a flexible bridge. Combine this increased number of gestures with more complex standard controls like those found on an Xbox One controller, and the possibilities are virtually endless.

Chapter 8. Bibliography

ADDIN Mendeley Bibliography CSL_BIBLIOGRAPHY Ahmaniemi, T. T., Kildal, J., &

Haveri, M. (2014). What is a device bend gesture really good for? In *Proceedings of the ACM conference on Human factors in computing systems* (pp. 3503–3512). New York, New York, USA, New York, USA: ACM Press. <http://doi.org/10.1145/2556288.2557306>

Arduino. (n.d.). <http://www.arduino.cc>.

Burstyn, J., Banerjee, A., & Vertegaal, R. (2012). FlexView: An Evaluation of Depth Navigation on Deformable Mobile Devices. In *Proc. TEI* (Vol. 1, pp. 193–200). Retrieved from <http://130.15.126.37/handle/1974/7450>

Casiez, G., Roussel, N., & Vogel, D. (2012). 1€ Filter: A Simple Speed-based Low-pass Filter for Noisy Input in Interactive Systems. *Proceedings of the 2012 ACM Annual Conference on Human Factors in Computing Systems - CHI '12*, 2527. <http://doi.org/10.1145/2207676.2208639>

Cassinelli, A., Angeseleva, J., Watanabe, Y., Frasca, G., & Ishikawa, M. (2012). Skin games. *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces - ITS '12*, 323. <http://doi.org/10.1145/2396636.2396690>

Chatham, A., Walmink, W., & Mueller, F. (2013). UnoJoy!: A Library for Rapid Video Game Prototyping Using Arduino. *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, 2787–2788. <http://doi.org/10.1145/2468356.2479512>

Daliri, F., & Girouard, A. (2016). Visual Feedforward Guides for Performing Bend Gestures on Deformable Prototypes. In *Graphics Interfaces 2016* (pp. 1–8).

Eady, A., & Girouard, A. (2015). Caret Manipulation using Deformable Input in Mobile Devices. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and*

- Embodied Interaction - TEI '15 - Work in Progress* (pp. 587–591). Stanford, CA: ACM, New York, NY, USA ©2015 . <http://doi.org/10.1145/2677199.2687916>
- Gallant, D. T., Seniuk, A. G., & Vertegaal, R. (2008). Towards more paper-like input. In *Proceedings of the 21st annual ACM symposium on User interface software and technology - UIST '08* (p. 283). New York, New York, USA. <http://doi.org/10.1145/1449715.1449762>
- Garcia-Sanjuan, F., Jaen, J., Catala, A., & Fitzpatrick, G. (2015). Airsteroids. *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces - ITS '15*, (November), 413–416. <http://doi.org/10.1145/2817721.2823480>
- Genyo, T. (1987). Mike Tyson's Punch-Out. Nintendo.
- Harmonix. (2005). Guitar Hero. Red Octane.
- Herkenrath, G., Karrer, T., & Borchers, J. (2008). Twend: Twisting and Bending as new Interaction Gesture in Mobile Devices. In *Proceeding of the twenty-sixth annual CHI conference extended abstracts on Human factors in computing systems* (p. 3819). New York, New York, USA. <http://doi.org/10.1145/1358628.1358936>
- Inoue, A. (2015). SHAPIO : Shape I / O Controller for Video Games, 565–570.
- Ionescu, D., Ionescu, B., Gadea, C., & Islam, S. (2011). A multimodal interaction method that combines gestures and physical game controllers. *Proceedings - International Conference on Computer Communications and Networks, ICCCN*, 1–6. <http://doi.org/10.1109/ICCCN.2011.6006085>
- Jones, B., Dillman, K., Aghel Manesh, S., Sharlin, E., & Tang, A. (2014). Designing an Immersive and Entertaining Pervasive Gameplay Experience with Sferos As Game and Interface Elements. *Proceedings of the First ACM SIGCHI Annual Symposium on Computer-Human Interaction in Play, 1*, 425–426. <http://doi.org/10.1145/2658537.2661301>

- JoyLabz LLC. (2016). Makey Makey. Retrieved July 30, 2016, from <http://www.makeymakey.com/>
- Kildal, J. (2012). Interacting with Deformable User Interfaces: Effect of Material Stiffness and Type of Deformation Gesture. In *Haptic and Audio Interaction Design* (pp. 71–80). http://doi.org/10.1007/978-3-642-32796-4_8
- Kildal, J., Lucero, A., & Boberg, M. (2013). Twisting Touch : Combining Deformation and Touch as Input within the Same Interaction Cycle on Handheld Devices. In *Proceedings of the 15th international conference on Human-computer interaction with mobile devices and services - MobileHCI '13* (p. 237). New York, New York, USA. <http://doi.org/10.1145/2493190.2493238>
- Kildal, J., Paasovaara, S., & Aaltonen, V. (2012). Kinetic device: Designing Interactions with a Deformable Mobile Interface. In *Proceedings of the ACM SIGCHI conference on Human Factors in Computing Systems Extended Abstracts* (p. 1871). <http://doi.org/10.1145/2212776.2223721>
- Konieczny, J., Shimizu, C., Meyer, G., & Colucci, D. (2005). A Handheld Flexible Display System. In *IEEE Visualization*. (pp. 591–597). IEEE. <http://doi.org/10.1109/VISUAL.2005.1532846>
- Lahey, B., Girouard, A., Burtleson, W., & Vertegaal, R. (2011). PaperPhone: Understanding the Use of Bend Gestures in Mobile Devices with Flexible Electronic Paper Displays. In *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11* (p. 1303). <http://doi.org/10.1145/1978942.1979136>
- Laugwitz, B., Held, T., & Schrepp, M. (2008). Construction and Evaluation of a User Experience Questionnaire. *HCI and Usability for Education and Work*, 63–76.

http://doi.org/10.1007/978-3-540-89350-9_6

Leap Motion Inc. (2016). Leap Motion. Retrieved August 29, 2016, from

<https://www.leapmotion.com/>

Lee, S.-S., Kim, S., Jin, B., Choi, E., Kim, B., Jia, X., ... Lee, K. (2010). How users manipulate deformable displays as input devices. In *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10* (p. 1647). New York, New York, USA.

<http://doi.org/10.1145/1753326.1753572>

Lee, S.-S., Maeng, S., Kim, D., Lee, K.-P., Lee, W., Kim, S., & Jung, S. (2011). FlexRemote:

Exploring the Effectiveness of Deformable User Interface as an Input Device for TV. In

HCI International - Poster's Extended Abstracts (pp. 62–65). [http://doi.org/10.1007/978-3-](http://doi.org/10.1007/978-3-642-22095-1_13)

[642-22095-1_13](http://doi.org/10.1007/978-3-642-22095-1_13)

Lo, J. (2013). *Bendy : An Exploration into Gaming with Mobile Flexible Devices*. Carleton University.

Maqsood, S., Chiasson, S., & Girouard, A. (2013). Passwords on flexible display devices. In

Proceedings of the ACM SIGSAC conference on Computer & communications security (pp.

1469–1472). <http://doi.org/10.1145/2508859.2512528>

Microsoft. (2010). Kinect.

Miyamoto, S. (1981). Donkey Kong. Nintendo. Retrieved from

[https://en.wikipedia.org/wiki/Donkey_Kong_\(video_game\)](https://en.wikipedia.org/wiki/Donkey_Kong_(video_game))

Miyamoto, S. (2014). Mario Kart 8. Nintendo.

Namco Bandai Games. (2015). neGcon Analogue Twist Controller. Retrieved August 2, 2016,

from [http://ridge-racer.tumblr.com/post/94244066204/negcon-analogue-twist-controller-](http://ridge-racer.tumblr.com/post/94244066204/negcon-analogue-twist-controller-this-unusual-if)
this-unusual-if

- Namco neGcon Controller. (1995). Retrieved July 26, 2016, from [http://www.computinghistory.org.uk/det/21581/Namco neGcon Controller/](http://www.computinghistory.org.uk/det/21581/Namco%20neGcon%20Controller/)
- Nguyen, V., Kumar, P., Yoon, S. H., Verma, A., & Ramani, K. (2015). SOFTii : Soft Tangible Interface for Continuous Control of Virtual Objects with Pressure-based Input. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction - TEI '15* (pp. 539–544). ACM Press. <http://doi.org/10.1145/2677199.2687898>
- Nguyen, V. P., Yoon, S. H., Verma, A., & Ramani, K. (2014). BendID: Flexible Interface for Localized Deformation Recognition. *UbiComp 2014*, 553–557. <http://doi.org/http://dx.doi.org/10.1145/2632048.2636092>
- Nintendo. (2006). Wii.
- OpenEmu. (2013). Retrieved from <http://openemu.org/>
- Pajitnov, A., & Pokhilko, V. (1984). Tetris. Nintendo.
- Priyadarshana, L. L., & Porter, V. (2016). MagicWand : Exploring Physical Affordances with a Handheld Cylindrical Display Object. *CHI Extended Abstracts on Human Factors in Computing Systems*, 3762–3765.
- Ramakers, R., Schöning, J., & Luyten, K. (2014). Paddle: Highly Deformable Mobile Devices with Physical Controls. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14* (pp. 2569–2578). New York, New York, USA: ACM Press. <http://doi.org/10.1145/2556288.2557340>
- Rendl, C., Haller, M., Izadi, S., Kim, D., Fanello, S., Parzer, P., ... Rothländer, T. (2014). FlexSense: a transparent self-sensing deformable surface. *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology - UIST '14*, 129–138. <http://doi.org/10.1145/2642918.2647405>

- Riyadh, M. (2014). Exploring Tapping with Thumb Input for Flexible Tablets. *CHI SRC*, 981–986. <http://doi.org/10.1145/2559206.2579422>
- Rovio Entertainment. (n.d.). Angry Birds.
- Schwesig, C., Poupyrev, I., & Mori, E. (2004). Gummi: A Bendable Computer. In *Proceedings of the 2004 conference on Human factors in computing systems - CHI '04* (pp. 263–270). <http://doi.org/10.1145/985692.985726>
- Shorey, P., Nguyen, Q., Dubord, N., Mukama, A., & Yang, S. (2014). O.R.B.S. Ottawa: Carleton University. Retrieved from <http://www.orbsgame.com>
- Siek, K. A., Rogers, Y., & Connelly, K. H. (2005). Fat finger worries: how older and younger users physically interact with PDAs. In *Human Computer Interaction - INTERACT* (pp. 267–280). Springer Berlin Heidelberg.
- Skalski, P., Tamborini, R., Shelton, a., Buncher, M., & Lindmark, P. (2011). Mapping the road to fun: Natural video game controllers, presence, and game enjoyment. *New Media & Society*, 13(2), 224–242. <http://doi.org/10.1177/1461444810370949>
- Smith, T. (2013). Squidge : An Integrated Game Controller, 2651–2654.
- Steimle, J., Jordt, A., & Maes, P. (2013). Flexpad : Highly Flexible Bending Interactions for Projected Handheld Displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13* (p. 237). ACM Press. <http://doi.org/10.1145/2470654.2470688>
- Strohmeier, P., Burstyn, J., Carrascal, J. P., Levesque, V., & Vertegaal, R. (2016). ReFlex: A Flexible Smartphone with Active Haptic Feedback for Bend Input. In *Proceedings of the TEI '16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction - TEI '16* (pp. 185–192). New York, New York, USA.

<http://doi.org/10.1145/2839462.2839494>

Tajika, T., Yonezawa, T., & Mitsunaga, N. (2008). Intuitive page-turning interface of e-books on flexible e-paper based on user studies. In *Proceeding of the 16th ACM international conference on Multimedia - MM '08* (p. 793). New York, New York, USA.

<http://doi.org/10.1145/1459359.1459489>

Tarun, A. P., Burleson, W., Environments, M., Lahey, B., Girouard, A., & Vertegaal, R. (2011). Snaplet : Using Body Shape to Inform Function in Mobile Flexible Display Devices, 329–334.

Troiano, G. M., Pedersen, E. W., & Hornbæk, K. (2015). Deformable Interfaces for Performing Music. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15* (pp. 377–386). New York, New York, USA: ACM Press.

<http://doi.org/10.1145/2702123.2702492>

Turn 10 Studios. (2005). Forza Motorsport. Microsoft Game Studios.

Villar, N., Gilleade, K. M., Ramdunyellis, D., & Gellersen, H. (2007). The VoodooIO gaming kit. *Computers in Entertainment*, 5(3), 7. <http://doi.org/10.1145/1316511.1316518>

Warren, K., Lo, J., Vadgama, V., & Girouard, A. (2013). Bending the rules: Bend Gesture Classification for Flexible Displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13* (p. 607). New York, New York, USA.

<http://doi.org/10.1145/2470654.2470740>

Watanabe, J., Mochizuki, A., & Horry, Y. (2008). Bookisheet: Bendable Device for Browsing Content Using the Metaphor of Leafing Through the Pages. In *Proceedings of the 10th international conference on Ubiquitous computing - UbiComp '08* (p. 360).

<http://doi.org/10.1145/1409635.1409684>

- Wigdor, D., & Wixon, D. (2011). *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture*. *Brave NUI World Designing Natural User Interfaces for Touch and Gesture*. <http://doi.org/10.1016/B978-0-12-382231-4.X0001-9>
- Wightman, D., Ginn, T., & Vertegaal, R. (2011). BendFlip: Examining Input Techniques for Electronic Book Readers with Flexible Form Factors. In *Proceeding INTERACT'11 Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction* (Vol. 6948, pp. 117–133). http://doi.org/10.1007/978-3-642-23765-2_9
- Ye, Z., & Khalid, H. (2010). Cobra: Flexible Displays for Mobile Gaming Scenarios. In *Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems - CHI EA '10* (p. 4363). New York, New York, USA. <http://doi.org/10.1145/1753846.1754154>

Chapter 9. Appendix

9.1. Consent Form

9.1.1. Experiment 1



CUREB clearance #: 15-286

Consent Form

Title: Continuous Flexible Input Methods in 3D Racing Games

Funding Source: NSERC Discovery Grant

Date of ethics clearance: December 16, 2015

Ethics Clearance for the Collection of Data Expires: August 31, 2016

I _____, choose to participate in a study on flexible input for video games. This study aims to determine if flexible input, combined with binary input, is a satisfying method of input when playing video games with a custom built controller. **The researcher for this study is Paden Shorey in the School of Information Technology.** He is working under the supervision of Audrey Girouard in the School for Information Technology.

This study takes approximately 60 minutes to complete and involves active participation. You will be asked to complete questionnaires before, after, and throughout the testing process. You will be asked to play a racing game using a custom made video game controller on two unique tracks. This game will be played using three different control schemes. There will be a total of six runs each followed up by a short user-experience questionnaire. With your consent, your participation will be recorded and kept safe for future analysis and observation. All data, including footage and stills from the footage, may be used in future publications and/or conference presentations/posters.

The minimal risk involved in participating in this study is comparable to an every day activity such as using a video game controller. You have the right to decline answering any open-ended questions or any questions you feel uncomfortable answering. You have the right to end your participation in the study at any time, for any reason, up until the end of the session. If you withdraw from the study, all information you have provided will be immediately destroyed.

You will be assigned a randomized ID at the beginning of your session so that your data will remain anonymous. With your consent, the video recording will focus on your hands and the controller and will not show your face. The researcher will leave the room when you are filling out any questionnaires to ensure that the responses remain anonymous.

Page 1 of 2

**This document has been printed on both sides of a single sheet of paper.
Please retain a copy of this document for your records.**

All research data will be saved in a private password-protected Dropbox accessible only by the researchers and research supervisor. Any hard copies of data (including any handwritten notes or questionnaires) will be kept in a locked office at Carleton University.

As a token of appreciation, you will receive a \$10 Tim Horton's gift card at the end of the session. This is yours to keep, even if you withdraw from the study.

Once the project is completed, all research data will remain in a private Dropbox and potentially used for other research projects on this same topic.

The ethics protocol for this project was reviewed by the Carleton University Research Ethics Board-B (CUREB-B), which provided clearance to carry out the research. Should you have questions or concerns related to your involvement in this research, please contact:

CUREB contact information:

Dr. Shelly Brown, Chair
Carleton University Research Ethics Board-B
Shelley.Brown@carleton.ca
613-520-2600 ext. 1505

You may also contact the Carleton University Research Office at ethics@carleton.ca.

Researcher contact information:

Paden Shorey
School of Information Technology
Carleton University
Tel: [REDACTED]
Email: paden.shorey@carleton.ca

Supervisor contact information:

Audrey Girouard
School of Information Technology
Carleton University
Tel: 613-520-2600 x8817
Email: audrey.girouard@carleton.ca

Do you agree to be video-recorded: ___Yes ___No

Signature of participant

Date

Signature of researcher

Date

9.1.2. Experiment 2



CUREB clearance #: 15-286

Consent Form

Title: Continuous Flexible Input Methods in 3D Racing Games

Funding Source: NSERC Discovery Grant

Date of ethics clearance: December 16, 2015

Ethics Clearance for the Collection of Data Expires: August 31, 2016

I _____, choose to participate in a study on flexible input for video games. This study aims to determine if flexible input, combined with binary input, is a satisfying method of input when playing video games with a custom built controller. **The researcher for this study is Paden Shorey in the School of Information Technology.** He is working under the supervision of Audrey Girouard in the School for Information Technology.

This study takes approximately 60 minutes to complete and involves active participation. You will be asked to complete questionnaires before, after, and throughout the testing process. You will be asked to play a racing game using a custom made video game controller on two unique tracks. This game will be played using three different control schemes. There will be a total of six runs each followed up by a short user-experience questionnaire. With your consent, your participation will be recorded and kept safe for future analysis and observation. All data, including footage and stills from the footage, may be used in future publications and/or conference presentations/posters.

The minimal risk involved in participating in this study is comparable to an every day activity such as using a video game controller. You have the right to decline answering any open-ended questions or any questions you feel uncomfortable answering. You have the right to end your participation in the study at any time, for any reason, up until the end of the session. If you withdraw from the study, all information you have provided will be immediately destroyed.

You will be assigned a randomized ID at the beginning of your session so that your data will remain anonymous. With your consent, the video recording will focus on your hands and the controller and will not show your face. The researcher will leave the room when you are filling out any questionnaires to ensure that the responses remain anonymous.

Page 1 of 2

**This document has been printed on both sides of a single sheet of paper.
Please retain a copy of this document for your records.**

All research data will be saved in a private password-protected Dropbox accessible only by the researchers and research supervisor. Any hard copies of data (including any handwritten notes or questionnaires) will be kept in a locked office at Carleton University.

As a token of appreciation, you will receive a \$10 Tim Horton's gift card at the end of the session. This is yours to keep, even if you withdraw from the study.

Once the project is completed, all research data will remain in a private Dropbox and potentially used for other research projects on this same topic.

The ethics protocol for this project was reviewed by the Carleton University Research Ethics Board-B (CUREB-B), which provided clearance to carry out the research. Should you have questions or concerns related to your involvement in this research, please contact:

CUREB contact information:

Dr. Shelly Brown, Chair
Carleton University Research Ethics Board-B
Shelley.Brown@carleton.ca
613-520-2600 ext. 1505

You may also contact the Carleton University Research Office at ethics@carleton.ca.

Researcher contact information:

Paden Shorey
School of Information Technology
Carleton University
Tel: [REDACTED]
Email: paden.shorey@carleton.ca

Supervisor contact information:

Audrey Girouard
School of Information Technology
Carleton University
Tel: 613-520-2600 x8817
Email: audrey.girouard@carleton.ca

Do you agree to be video-recorded: ___Yes ___No

Signature of participant

Date

Signature of researcher

Date

9.2. Questionnaires

9.2.1. Experiment 1

7/18/2016

LimeSurvey - Flexible Input Methods for Classic Video Games

Flexible Input Methods for Classic Video Games

Questionnaires for the first study.

Hello.

Please fill out the following questions as accurately as possible.

There are 19 questions in this survey

Pre Questionnaire

[]

Participant ID: *

Please write your answer here:

This is to be filled out by the study supervisor.

[]1: Age *

Please write your answer here:

[]2: Sex *

Please choose **only one** of the following:

- Female
- Male

[]3: Handedness *

Please choose **only one** of the following:

- Right-Handed
- Left-Handed
- Ambidextrous

[[4: How often do you play video games? *

Please choose **only one** of the following:

- < 1 hour weekly
- 1 - 3 hours weekly
- 3- 10 hours weekly
- > 10 hours weekly

[[5: Which types of games do you play on a weekly basis (please select all that apply)? *

Please choose **all that apply**:

- Console (Xbox One, Playstation 4, Wii U, etc.)
- Mobile (iPhone, Android, etc.)
- PC/Mac
- Social Media (Facebook)
- None
- Other:

[[6: Have you ever used a flexible input method? If so, where and in what context? *

Please choose **only one** of the following:

- Yes
- No

Make a comment on your choice here:

[]7: Have you ever played a game using a non-standard form of input (ie. guitar peripheral, skateboard, Leap Motion, etc.)? If so, please state what you used. *

Please choose **only one** of the following:

- Yes
 No

Make a comment on your choice here:

[]Please rate your experience level with the following games (1 = No Experience, 5 = Lots of Experience).

[]8: Donkey Kong (Nintendo Entertainment System) *

Please choose **only one** of the following:

- 1
 2
 3
 4
 5

[]9: Mike Tyson's Punch-Out (Nintendo Entertainment System) *

Please choose **only one** of the following:

- 1
- 2
- 3
- 4
- 5

[]10: Tetris (Nintendo Entertainment System) *

Please choose **only one** of the following:

- 1
- 2
- 3
- 4
- 5

Post Questionnaire

[]

Please rate the following statements from 1 (Strongly Disagree) to 5 (Strongly Agree) based on your experience with the flexible game controller.

[[1]: The controller was comfortable to use. *

Please choose **only one** of the following:

- 1
- 2
- 3
- 4
- 5

[[2]: The controller was fun to use. *

Please choose **only one** of the following:

- 1
- 2
- 3
- 4
- 5

[[3]: I would use this controller to play other games. *

Please choose **only one** of the following:

- 1
- 2
- 3
- 4
- 5

[[4: Which game did you have the most fun playing with the bend controls? Why? *

Please choose **only one** of the following:

- Donkey Kong
- Mike Tyson's Punch-Out
- Tetris

Make a comment on your choice here:

[[5: Overall, how would you describe your experience with the bend input?

Please write your answer here:

[]6: If any, what other games would you like to play using bend controls?

Please write your answer here:

NASA TLX & UEQ - Study 1

There are 5 questions in this survey

NASA TLX & UEQ

[] Participant ID: *

Please write your answer here:

This will be fill out by the research supervisor.

[] Game: *

Please choose the appropriate response for each item:

| | | | |
|-------------|-----------------------|------------------------|-----------------------|
| Game Played | Donkey Kong | Mike Tyson's Punch-Out | Tetris |
| | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

[] Control Scheme: *

Please choose the appropriate response for each item:

| | | | |
|---------------------|-----------------------|-----------------------|-----------------------|
| Control Scheme Used | Binary (1) | Navigation (2) | Action (3) |
| | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

[] *

Please choose the appropriate response for each item:

| | 1 (Very Low) | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 (Very High) |
|--|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| How mentally demanding was the task? | <input type="radio"/> |
| How physically demanding was the task? | <input type="radio"/> |
| How hurried or rushed was the pace of the task? | <input type="radio"/> |
| How successful were you in accomplishing what you were asked to do? | <input type="radio"/> |
| How hard did you have to work to accomplish your level of performance? | <input type="radio"/> |
| How insecure, discouraged, irritated, stressed, and annoyed were you? | <input type="radio"/> |

[]

For the assessment of the product, please fill out the following questionnaire. The questionnaire consists of pairs of contrasting attributes that may apply to the product. The circles between the attributes represent gradations between the opposites. You can express your agreement with the attributes by ticking the circle that most closely reflects your impression.

Please decide spontaneously. Don't think too long about your decision to make sure that you convey your original impression.

It is your personal opinion that counts. Please remember: there is no wrong or right answer!

*

Please choose the appropriate response for each item:

| | 1 | 2 | 3 | 4 | 5 |
|--|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| annoying -> enjoyable | <input type="radio"/> |
| not understandable -> understandable | <input type="radio"/> |
| creative -> dull | <input type="radio"/> |
| easy to learn -> difficult to learn | <input type="radio"/> |
| valuable -> inferior | <input type="radio"/> |
| boring -> exciting | <input type="radio"/> |
| not interesting -> interesting | <input type="radio"/> |
| unpredictable -> predictable | <input type="radio"/> |
| fast -> slow | <input type="radio"/> |
| inventive -> conventional | <input type="radio"/> |
| obstructive -> supportive | <input type="radio"/> |
| good -> bad | <input type="radio"/> |
| complicated -> easy | <input type="radio"/> |
| unlikable -> pleasing | <input type="radio"/> |
| usual -> leading edge | <input type="radio"/> |
| unpleasant -> pleasant | <input type="radio"/> |
| secure -> not secure | <input type="radio"/> |
| motivating -> demotivating | <input type="radio"/> |
| meets expectations -> does not meet expectations | <input type="radio"/> |
| inefficient -> efficient | <input type="radio"/> |
| clear -> confusing | <input type="radio"/> |
| impractical -> practical | <input type="radio"/> |
| organized -> cluttered | <input type="radio"/> |
| attractive -> unattractive | <input type="radio"/> |
| friendly -> unfriendly | <input type="radio"/> |
| conservative -> innovative | <input type="radio"/> |

Game Experience Survey

There are 7 questions in this survey

Game Questions

[] Participant ID: *

Please write your answer here:

This will be filled out by the research supervisor.

[] Game Played *

Please choose **only one** of the following:

- Donkey Kong
- Mike Tyson's Punch-Out
- Tetris

[] Please rank the following input methods based on the given category as it applies to the last game you played.

[] 1: Most Fun *

Please number each box in order of preference from 1 to 3

- Just Buttons
- Bend/Twist as Navigation
- Bend/Twist as Action

[[2: Most Natural *

Please number each box in order of preference from 1 to 3

| | |
|----------------------|--------------------------|
| <input type="text"/> | Just Buttons |
| <input type="text"/> | Bend/Twist as Navigation |
| <input type="text"/> | Bend/Twist as Action |

[[3: Most Frustrating *

Please number each box in order of preference from 1 to 3

| | |
|----------------------|--------------------------|
| <input type="text"/> | Just Buttons |
| <input type="text"/> | Bend/Twist as Navigation |
| <input type="text"/> | Bend/Twist as Action |

[[4: Most Comfortable *

Please number each box in order of preference from 1 to 3

| | |
|----------------------|--------------------------|
| <input type="text"/> | Just Buttons |
| <input type="text"/> | Bend/Twist as Navigation |
| <input type="text"/> | Bend/Twist as Action |

9.2.2. Experiment 2

7/19/2016

LimeSurvey - Study 2

Study 2

There are 21 questions in this survey

Pre-Questionnaire

[] Participant ID *

Please write your answer here:

The research supervisor will fill this in.

[] Age *

Please write your answer here:

[] Sex *

Please choose **only one** of the following:

- Male
 Female

[] Handedness *

Please choose **only one** of the following:

- Right Handed
 Left Handed
 Ambidextrous

[] How often do you play video games? *

Please choose **only one** of the following:

- Never
 Occasionally
 Frequently

[] Have you ever used a flexible input method (bending or twisting a device in order to interact with it)? *

Please choose **only one** of the following:

- Yes
 No

[] Have you previously used this prototype? *

Please choose **only one** of the following:

- Yes
 No

[]How would you rate your EXPERIENCE with racing games (1 = No Experience, 5 = Lots of Experience)? *

Please choose **only one** of the following:

- 1
- 2
- 3
- 4
- 5

[]How would you rate your SKILL with racing games (1 = No Skill, 5 = Lots of Skill)? *

Please choose **only one** of the following:

- 1
- 2
- 3
- 4
- 5

Post-Questionnaire

[]

BINARY **ACTION** **NAVIGATION**

SPEED **PRECISION**

[] How easy was it to control your racer using following scheme/track combinations? (1 = Very Hard, 5 = Very Easy) *

Please choose the appropriate response for each item:

| | 1 | 2 | 3 | 4 | 5 |
|----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| Binary/Speed | <input type="radio"/> |
| Action/Speed | <input type="radio"/> |
| Navigation/Speed | <input type="radio"/> |
| Binary/Precision | <input type="radio"/> |
| Action/Precision | <input type="radio"/> |
| Navigation/Precision | <input type="radio"/> |

[] For the speed track, rank the schemes by preference (most preferred to least preferred). Explain your choice. *

Please number each box in order of preference from 1 to 3

Binary

Action

Navigation

[] *

Please write your answer here:

[] * For the precision track, rank the schemes by preference (most preferred to least preferred). Explain your choice. *

Please number each box in order of preference from 1 to 3

| | |
|----------------------|------------|
| <input type="text"/> | Binary |
| <input type="text"/> | Action |
| <input type="text"/> | Navigation |

[] *

Please write your answer here:

[] Which track was better suited for the binary scheme? Why? *

Please choose **only one** of the following:

- Speed
 Precision

Make a comment on your choice here:

[] Which track was better suited for the action scheme? Why? *

Please choose **only one** of the following:

- Speed
 Precision

Make a comment on your choice here:

[] Which track was better suited for the navigation scheme? Why? *Please choose **only one** of the following:

- Speed
 Precision

Make a comment on your choice here:

[] Overall, which scheme/track combination did you have the most fun playing? *Please choose **only one** of the following:

- Binary/Speed
 Action/Speed
 Navigation/Speed
 Binary/Precision
 Action/Precision
 Navigation/Precision

[] Overall, which scheme/track combination did you have the least fun playing? *Please choose **only one** of the following:

- Binary/Speed
 Action/Speed
 Navigation/Speed
 Binary/Precision
 Action/Precision
 Navigation/Precision

[] How would you describe your overall experience? *

Please write your answer here:

UEQ - Study 2

There are 4 questions in this survey

UEQ

[]Participant ID *

Please write your answer here:

[]Scheme *

Please choose **only one** of the following:

- Binary
- Action
- Navigation

[]Track *

Please choose **only one** of the following:

- Speed
- Precision

[] For the assessment of the product, please fill out the following questionnaire. The questionnaire consists of pairs of contrasting attributes that may apply to the product. The circles between the attributes represent gradations between the opposites. You can express your agreement with the attributes by ticking the circle that most closely reflects your impression.

Please decide spontaneously. Don't think too long about your decision to make sure that you convey your original impression.

It is your personal opinion that counts. Please remember: there is no wrong or right answer! *

Please choose the appropriate response for each item:

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|--|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| annoying -> enjoyable | <input type="radio"/> |
| not understandable -> understandable | <input type="radio"/> |
| creative -> dull | <input type="radio"/> |
| easy to learn -> difficult to learn | <input type="radio"/> |
| valuable -> inferior | <input type="radio"/> |
| boring -> exciting | <input type="radio"/> |
| not interesting -> interesting | <input type="radio"/> |
| unpredictable -> predictable | <input type="radio"/> |
| fast -> slow | <input type="radio"/> |
| inventive -> conventional | <input type="radio"/> |
| obstructive -> supportive | <input type="radio"/> |
| good -> bad | <input type="radio"/> |
| complicated -> easy | <input type="radio"/> |
| unlikable -> pleasing | <input type="radio"/> |
| usual -> leading edge | <input type="radio"/> |
| unpleasant -> pleasant | <input type="radio"/> |
| secure -> not secure | <input type="radio"/> |
| motivating -> demotivating | <input type="radio"/> |
| meets expectations -> does not meet expectations | <input type="radio"/> |
| inefficient -> efficient | <input type="radio"/> |
| clear -> confusing | <input type="radio"/> |
| impractical -> practical | <input type="radio"/> |
| organized -> cluttered | <input type="radio"/> |
| attractive -> unattractive | <input type="radio"/> |
| friendly -> unfriendly | <input type="radio"/> |
| conservative -> innovative | <input type="radio"/> |

Post-Trial Questionnaire

There are 7 questions in this survey

Questions

[] Participant ID *

Please write your answer here:

[] Scheme *

Please choose **only one** of the following:

- Binary
- Action
- Navigation

[] Track *

Please choose **only one** of the following:

- Speed
- Precision

[] How well do you think you performed? (1=Very Poorly, 5=Very Well) *

Please choose **only one** of the following:

- 1
- 2
- 3
- 4
- 5

[]How natural was this control scheme for this track? (1=Very Unnatural, 5=Very Natural) *

Please choose **only one** of the following:

- 1
- 2
- 3
- 4
- 5

[]How fun was this control scheme for this track? (1=Very Boring, 5=Very Fun) *

Please choose **only one** of the following:

- 1
- 2
- 3
- 4
- 5

[]How frustrating was this control scheme for this track? (1=Not Frustrating, 5=Very Frustrating) *

Please choose **only one** of the following:

- 1
- 2
- 3
- 4
- 5