

IMPROVING THE ACCESSIBILITY OF DIGITAL GAMES USING  
PARTIAL AUTOMATION

by

GABRIELE CIMOLINO

A thesis submitted to the School of Computing  
in conformity with the requirements for the  
Degree of Doctor of Philosophy

Queen's University

Kingston, Ontario, Canada

December 2023

Copyright © Gabriele Cimolino, 2023

---

## Abstract

---

PARTIAL automation is a games accessibility technique that can improve the accessibility of digital games to novices and players with disabilities. Under partial automation, the player shares control of the game with an AI copilot that performs gameplay actions that the player has difficulty controlling. Through a series of studies, we found that this technique can extend a game's accessibility to players who might be unable to play otherwise. This thesis contributes an exploration of the design space of partial automation, including many examples of partial automation created for use in our studies, as well as empirical evidence that partial automation can improve a game's accessibility.

---

## Dedication

---

*This thesis is dedicated to the people who supported its creation. It is dedicated to my supervisor Nick, who always leads me to victory, and the equites of the EQUIS Lab; to Sara and the iStudio Lab; to the Queen's School of Computing; to Andrea and the Kingston Peer Connections community; to my co-authors Sussan, Carl, and Renee; to my family and to my fiancée, Kayleigh.*

---

## Co-authorship

---

The novel contributions presented in this thesis (Section 1.2) are our own and were created with the help of our collaborators. The research reported in Chapter 6 was conducted in collaboration with Dr. Sussan Askari who gave feedback on the study protocol, recruited participants, helped with study management, and gave editorial comments on the papers published from this work. The research reported in Chapter 7 was conducted with the help of Dr. Carl Gutwin, who contributed to the study design and provided feedback on the text of the paper published from this work, as well as Renee (Xinyu) Chen, who assisted in developing the study protocol.

# Table of Contents

<b>Abstract</b>	i
<b>Dedication</b>	ii
<b>Co-authorship</b>	iii
<b>List of Tables</b>	vi
<b>List of Figures</b>	viii
<b>1 Introduction</b>	1
1.1 Thesis Overview . . . . .	4
1.2 Thesis Contributions . . . . .	6
<b>I Background</b>	8
<b>2 Motivation</b>	9
2.1 Gameplay Barriers Facing Disabled Players . . . . .	10
2.2 The Uses of Digital Games . . . . .	16
2.3 Summary . . . . .	24
<b>3 Games Accessibility &amp; Partial Automation</b>	25
3.1 Approaches to Designing Accessible Games . . . . .	26
3.2 Combining Approaches . . . . .	36
3.3 Partial Automation . . . . .	38
3.4 Ontology of Approaches to Games Accessibility . . . . .	43
3.5 Summary . . . . .	44
<b>4 Human-AI Shared Control &amp; Human-Machine Cooperation</b>	46
4.1 Human-AI Shared Control . . . . .	47
4.2 Human-Machine Cooperation . . . . .	61

4.3	Summary . . . . .	67
<b>II</b>	<b>Research in Partial Automation</b>	<b>69</b>
5	<b>Shared Control Design Space Analysis</b>	<b>70</b>
5.1	Shared Control Design . . . . .	72
5.2	Dimension Space . . . . .	76
5.3	Shared Control Design Patterns . . . . .	90
5.4	Generative Shared Control Design . . . . .	105
5.5	Summary . . . . .	110
6	<b>Personalizable Partial Automation</b>	<b>112</b>
6.1	Study Design . . . . .	114
6.2	Results: Ability to Play . . . . .	131
6.3	Results: Player Experience . . . . .	133
6.4	Design Implications for Partial Automation . . . . .	153
6.5	Summary . . . . .	159
7	<b>Automation Confusion</b>	<b>161</b>
7.1	Study Design . . . . .	164
7.2	Results . . . . .	175
7.3	Design Implications for Partial Automation . . . . .	197
7.4	Summary . . . . .	200
<b>III</b>	<b>Discussion &amp; Conclusion</b>	<b>202</b>
8	<b>Partial Automation Architecture &amp; Toolkit</b>	<b>203</b>
8.1	The Concordia Architecture . . . . .	205
8.2	The Concordia Toolkit . . . . .	217
8.3	Summary . . . . .	228
9	<b>Conclusion</b>	<b>229</b>
9.1	Limitations . . . . .	231
9.2	Future Work . . . . .	234
9.3	Summary . . . . .	237
A	Health Sciences Research Ethics Board (HSREB) Approval . . . . .	295
B	General Research Ethics Board (GREB) Approval . . . . .	298

# List of Tables

5.1	The systems examined, topics encountered, and search terms used in each phase of our survey. . . . .	77
5.2	Our corpus of shared control systems according to their classifications in our dimension space. More intense colors indicate greater extension along an axis. <i>Italicized</i> names are those designated by the system's designers. . . . .	78
6.1	Participants' demographic information. AIS grades indicate whether their injury is complete, meaning no sensory or motor function was preserved at sacral segments S4-S5, or incomplete, meaning some sensory or motor function was retained. NLI (Neurological Level of Injury) specifies the part of the spine that was injured. . . . .	126
6.2	Participants' gaming information. $GF_{<}$ and $GF_{>}$ denote participants' gaming frequency before and after injury respectively. The Cycling column indicates whether participants actively or passively cycled during play. The Movement, Direction, and Action columns indicate the devices participants used to control each input ("V"=viva2, "G"=gamepad, "J"=joystick, "B"=bite switch; no device indicator means input was automated). . . . .	127
6.3	Players abilities to carry out gameplay actions in the rehabilitation games. A ✓ indicates that the participant performed the action at least once during gameplay. A ~ indicates that an error prevented gameplay data collection. . . . .	131
7.1	Each game's awareness cues and their meanings. . . . .	170

7.2 Participants' demographic data including the game genres they had played before (Genres), the number of hours they spent playing games each week (Current Gaming H/W), the number of hours they spent playing games each week at their peak (Peak Gaming H/W), whether they mentioned wanting to play games with friends or siblings (Friend), and whether they mentioned wanting to play games with their children (Parent). Participants' mean age was 38.8 and their mode gender was woman. . . . .	171
7.3 Definitions for all of automation confusion's concepts. . . . .	174
7.4 Definitions for the types of mental model errors experienced by participants. . . . .	180
7.5 Definitions for the attitudes towards the games experienced by participants. . . . .	184
7.6 Definitions for the behaviours resulting from confusion exhibited by participants. . . . .	188
7.7 Definitions for the sources of confusion experienced by participants. .	192

# List of Figures

2.1	The game interaction model of Yuan et al. (329). . . . .	10
2.2	Example game controllers. . . . .	12
2.3	A child playing <i>Wii Sports</i> and the controller used to play it. . . . .	13
2.4	The first video game console and one of its games. . . . .	17
2.5	The <i>Skyfarer</i> shoulder mobility game. . . . .	22
3.1	Children playing <i>Liberi</i> . . . . .	27
3.2	One-switch play. . . . .	28
3.3	People with different types of mobility aids playing <i>Liberi</i> . . . . .	29
3.4	Games with focused accessibility. . . . .	31
3.5	People playing active games. . . . .	32
3.6	Adapted interfaces. . . . .	34
3.7	People playing <i>Wheelchair Revolution</i> (102) using the default dance mat controller (left) and the alternative KINECT <sup>Wheels</sup> controller (right). . . . .	37
3.8	Two examples of input automation (left) and one-switch automation (right) in <i>Space Invaders</i> . The game inputs in <i>Space Invaders</i> are called <i>shoot</i> , which makes the avatar shoot, and <i>move</i> , which makes the avatar move left and right. . . . .	41
3.9	A novel ontology of approaches to games accessibility and relations between them. <i>Extends</i> means that one approach overcomes the limitations of another and <i>complements</i> means that both approaches overcome each other's limitations. For example, as explained in Section 3.2, each of interface adaptation and player balancing <i>overcome the limitations of</i> the other, but neither interface adaptation nor player balancing <i>overcome the limitations of</i> partial automation. . . . .	44

4.1	Norman's seven stages of action grounded in the game interaction model introduced in Chapter 2. After forming a <i>goal</i> , users <i>plan</i> how they will achieve it and <i>specify</i> the sequence of actions needed to enact their plans. Users then <i>perform</i> the action sequence and <i>perceive</i> how their actions affect the state of the world. Finally, users <i>interpret</i> their perceptions and <i>compare</i> their interpretations to their goals to determine whether they were achieved. The <i>perceive</i> stage corresponds to <i>receive stimuli</i> . The <i>interpret</i> , <i>compare</i> , <i>goal</i> , <i>plan</i> , and <i>specify</i> stages correspond to <i>determine responses</i> . The <i>perform</i> stage corresponds to <i>provide input</i> . . . . .	62
5.1	Visual representations of our dimension space. . . . .	79
5.2	The two design patterns identified in digital games. Although both are forms of partial automation, we use the term <i>player balancing</i> to distinguish partial automation that amends players control to assist with shared tasks such as aiming and steering. . . . .	93
5.3	The two design patterns identified in semi-autonomous driving. . . . .	95
5.4	The design pattern identified in creativity support. . . . .	96
5.5	The design pattern identified in mobility assistance. . . . .	97
5.6	The design pattern identified in telerobotics. . . . .	98
5.7	The design pattern identified in surgery. . . . .	99
5.8	The <i>data view</i> RadViz plot generated by the <i>Subspaces Explorer</i> tool and higher-level design patterns used across multiple domains. In the data view in the center, each of the systems in our corpus are represented as circles colored according to their domains and plotted according to their similarities with other systems as well as the axes along which they are similar. For example, the cluster in the top right corner use a distinctive combination of Influence and Mediation, while the cluster at the bottom places the AI in a unique AI Role. Systems found in the center of the plot did not exhibit a cohesive pattern, so this cluster was not selected. Selected patterns are encased in colored rectangles that each correspond to a diagram of the same color on the right and left. . . . .	100
5.9	The <i>vigilant savior</i> design pattern. . . . .	102
5.10	The <i>supportive patron</i> design pattern. . . . .	103
5.11	The <i>compromise negotiator</i> design pattern. . . . .	104
5.12	The <i>equal partner</i> design pattern. . . . .	105
5.13	The Kiviat diagram representing our novel semi-autonomous driving system. . . . .	107
5.14	The Kiviat diagram representing our novel creativity support system.	108

5.15	The Kiviat diagram representing our novel telerobotics system. . . . .	109
6.1	A cycling device commonly used in spinal cord injury rehabilitation. .	115
6.2	The two partially automated exergames used in the study. . . . .	117
6.3	A man with complete paraplegia playing <i>Dozo Quest</i> using the gamepad to provide Direction and Action inputs. The viva2 is pedalling, and so copilot provides Movement input. . . . .	118
6.4	The devices used by participants. . . . .	120
6.5	A woman with incomplete tetraplegia playing <i>Dino Dash</i> using a joystick and bite switch to provide Direction and Action inputs. The viva2 is pedalling for her, and so the copilot provides Movement input.	121
6.6	A man using a MOTOMed viva2 (left of image) for rehabilitation of spinal cord injury. The viva2 is a pedalling device where the users' feet are secured to the device while they sit in their own wheelchairs. The device's motor moves the pedals for users who are unable to pedal for themselves. . . . .	122
6.7	Partial automation: The game interface offers inputs for controlling movement, direction, and game actions. The player controls as many of these inputs as they can (in this example, Action and Direction). An AI copilot provides inputs for parts of the interface that the player cannot control (in this example, Movement). . . . .	123
6.8	The MOTOMax game shipped with the viva2 rewards players for symmetric pedalling. . . . .	128
7.1	Screenshots from <i>Ninja Showdown</i> 's tutorial. In the image on the top left, Emi turns her head to look at the player, indicating that she is awaiting player input. In the bottom left, she displays an Intention cue indicating that she intends to use a Bomb if the player does not tell her to use the Sword. In the image on the bottom right, Emi has initiated the Bomb attack and displays an Action cue, indicating that she decided to use the Bomb. . . . .	166
7.2	Screenshots from <i>Spelunky</i> 's tutorial. In each image, an explanation for how to make the avatar perform an action, as well as the action's corresponding Option cue, are displayed. Refusal cues were purely auditory and so are not visualized. . . . .	168
7.3	The analytical process we followed during theoretical development. .	176
7.4	Storyboard depicting P2 pressing the 'b' key to make Emi use the Bomb.	178
7.5	Storyboard depicting P7 mashing the spacebar because she felt uninvolved in play. . . . .	178
7.6	Storyboard depicting P3 pressing on the left and right sides of the spacebar to guide her avatar. . . . .	179

8.1	The Concordia abstract architecture. Arrows represent data flow between modules. Data flow related to providing input to the game is red, while data flow related to the game's state are blue. . . . .	210
8.2	The Concordia reference architecture. Concordia requires an interface exposing the game's state and provides interfaces exposing awareness information to be presented as awareness cues as well as merged game inputs from the player and the copilot. Each of the Concordia architecture's components, as well as the interfaces they provide and require, are shown. The plug-ins that each component uses are coloured according to the situations in which they need to be modified. Blue plug-ins change in response to changes in the game. Green plug-ins change in response to changes in the types of inputs. Red plug-ins change in response to changes in the types of inputs or the runtime environment. . . . .	213
8.3	<i>Interface Invaders.</i> The space ship at the bottom of the screen is the player's avatar and the shapes above are the enemies. The red box around the enemy on the left is an awareness cue indicating which enemy the copilot has chosen to target. The red line below the player's avatar is an awareness cue indicating that the copilot is avoiding the laser above. . . . .	218
8.4	The Concordia settings menu for <i>Interface Invaders</i> . In this example, the player controls the Shoot input using a bite switch while the copilot controls the Move input. . . . .	221

# CHAPTER 1

---

## Introduction

---

THIS thesis proposes and investigates partial automation as a games accessibility technique that can assist players who have difficulty controlling digital games. In partial automation, players share control with an AI copilot that controls those parts of the game that the player cannot. For novices and people with disabilities, there may be multiple aspects of play that make controlling a digital game difficult. Players need to quickly make sense of what is happening in the game, decide what to do about it, and manipulate a gaming controller to make that happen. But novices may not yet be capable of making gameplay decisions quickly and may not yet be proficient in using gaming controllers. Players with physical disabilities may have difficulty manipulating controllers quickly enough to keep up with the game, they may be unable to use some parts of controllers, or

they may be unable to use any compatible controllers. Overall, providing input to games can be overly challenging for some players and may render them unable to play. Although there are several existing approaches to accessible game design that can make games easier to control, the efficacy of these approaches has limits. In this thesis, we propose and evaluate a new approach that can extend a game’s accessibility further by dividing control of the game between the human player and an AI agent.

In partial automation we say that games have *inputs* that players control to make things happen in the game. For example, the game *Space Invaders* [289] has a *Shoot* input, which makes the player’s avatar shoot when the player presses a button, and a *Move* input, which makes the player’s avatar move horizontally when the player tilts a joystick. Partial automation provides a *copilot AI*—an intelligent partner such as in GitHub Copilot<sup>1</sup> or Microsoft Copilot<sup>2</sup>—that shares control of the game with the player. For example, if a *Space Invaders* player is able to press a button but cannot use a joystick, then partial automation can control the *Move* input while the player controls the *Shoot* input. This approach may improve a game’s accessibility to players who have difficulty deciding what input to provide, who have difficulty manipulating a controller precisely enough to make the game do what they want, or who cannot use some parts of the game’s controller.

Although there are many interactive systems that employ the *human-AI shared control* interaction paradigm, in which a human user and an AI agent cooperatively control a system, it is not yet known whether partial automation can improve the accessibility of digital games. Several existing games have accessibility features

---

<sup>1</sup>[GitHub Copilot](#)

<sup>2</sup>[Microsoft Copilot](#)

that could be called partial automation, but the efficacy of this approach has not been investigated systematically. For example, *Mario Kart 8 Deluxe* [219] provides ‘Smart Steering’ and ‘Auto-accelerate’ features that putatively improve the game’s accessibility<sup>3</sup>. However, it is not yet known whether these features can extend the game’s accessibility, how using these features might affect players’ experiences of play, or what new problems might arise.

This thesis is intended to establish the fundamental design knowledge needed to improve the accessibility of digital games using partial automation. We relate partial automation to other uses of human-AI shared control and demonstrate partial automation’s capacity to extend a game’s accessibility to non-gamers<sup>4</sup> and players with motor impairments. Through an in-hospital study in which patients played rehabilitation games with partial automation, we learned that partial automation can help disabled players to play games that they are unable to play otherwise. Participants liked playing rehabilitation games with partial automation and highly valued being able to personalize the assistance afforded by partial automation as well as the hardware interfaces they used to play. Through an in-lab study in which non-gamers played fast-paced action games with partial automation, we learned that partial automation can make players confused about how games are controlled if players’ understanding and awareness of the copilot are not properly supported. Players may experience *automation confusion* and misattribute the copilot’s actions to themselves when they cannot make the game do what they want. These findings have informed the design of a reference software architecture and an associated development toolkit intended to facilitate the creation of partial automation in a

---

<sup>3</sup>The Family Gaming Database’s accessibility report for *Mario Kart 8 Deluxe*

<sup>4</sup>We define non-gamers as people who do not play games for their own enjoyment.

typical game development process. Ultimately, this thesis serves to substantiate the following thesis statement:

Partial automation can improve the accessibility of digital games to non-gamers and players with motor impairments.

## 1.1 Thesis Overview

This thesis is organized into three parts composed of multiple chapters. Part 1: [Background](#) establishes the context and motivation for the work described in subsequent parts. It introduces two groups of players who encounter accessibility barriers in games (i.e., non-gamers & players with motor impairments), explains how some of these barriers have previously been addressed by existing approaches to accessible game design, and describes how human-AI shared control has been used to improve users' interactions with other interactive systems. Chapter 2: [Motivation](#) describes the social and medical contexts in which digital games are played and argues for the necessity of games accessibility. Chapter 3: [Games Accessibility & Partial Automation](#) describes fundamental problems of games accessibility and explains, through illustrative examples drawn from the literature, the various approaches designers can take to improve the accessibility of games. This chapter also introduces the notion of partial automation and explains its potential benefits with reference to the limitations of other approaches to accessible game design. Chapter 4: [Human-AI Shared Control & Human-Machine Cooperation](#) introduces the notion of human-AI shared control and describes some of the ways in which

interactive systems have been designed for human users to share control with AI agents. This chapter also discusses common issues of human-machine cooperation and explain how they relate to players' interactions with partial automation.

Part 2: [Research in Partial Automation](#) presents our novel research on partial automation. It contains an analysis of the human-AI shared control design space and two studies in which players with motor impairments and non-gamers played games with partial automation. Chapter 5: [Shared Control Design Space Analysis](#) describes uses of human-AI shared control in six problem domains and proposes a dimension space intended to guide the design of human-AI shared control systems. This chapter also discusses a design space analysis to elucidate the most common design patterns used across multiple domains. The research question posed in this chapter is **RQ1**: “*What are the dimensions for design and design patterns supporting human-AI shared control?*” The research reported in this chapter was published at the *ACM CHI Conference on Human Factors in Computing Systems* [54]. Chapter 6: [Personalizable Partial Automation](#) describes a personalizable form of partial automation and presents an in-hospital study in which partial automation made rehabilitation games accessible to patients with different abilities due to spinal cord injury. This chapter also discusses partial automation’s capacity to broaden a game’s accessibility as well as the themes generated through a thematic analysis of participants’ feedback. The research question posed in this chapter is **RQ2**: “*Does personalizable partial automation make games accessible to players with radically different motor abilities while remaining fun to play?*” The research reported in this chapter was published in the *Proceedings of the Annual Symposium on Computer-Human Interaction in Play* [52] and in the *Proceedings of the 23rd International ACM*

*SIGACCESS Conference on Computers and Accessibility* [51]. Chapter 7: [Automation Confusion](#) presents an in-lab study of non-gamers playing partially automated one-button action games. This chapter also presents a grounded theory of participants' automation confusion and demonstrates how automation confusion occurs through examples drawn from our analyses. The research question posed in this chapter is **RQ3**: “*What are the sources and types of confusion engendered by partial automation?*” The research reported in this chapter was published at the *ACM CHI Conference on Human Factors in Computing Systems* [53].

Part 3: [Discussion & Conclusion](#) summarizes our findings and their potential implications for the design and development of partial automation. Chapter 8: [Partial Automation Architecture & Toolkit](#) proposes a novel reference architecture for partial automation intended to facilitate the iterative refinement of partial automation in a typical game development process. This chapter also presents an associated development toolkit and demonstrates its use in a game implemented using the Unity game engine. Chapter 9: [Conclusion](#) summarizes our findings, discusses the limitations of our studies, and proposes several promising directions for future work.

## 1.2 Thesis Contributions

The major contribution of this thesis is partial automation itself, which has gone unnamed and uninvestigated until now. The chapters of this thesis make numerous contributions in service of exploring the design space of partial automation, understanding its capacity to improve games accessibility, and facilitating its use in digital games. Below, we enumerate these novel contributions in the order of their appearance in subsequent chapters.

**Chapter 3: Games Accessibility & Partial Automation** An ontology of approaches to accessible game design (Section 3.4)

**Chapter 4: Human-AI Shared Control & Human-Machine Cooperation** A scoping review of human-AI shared control systems and application domains (Section 4.1)

**Chapter 5: Shared Control Design Space Analysis** A dimension space of human-AI shared control designs (Section 5.2) and an analysis of shared control design patterns used within and across application domains (Section 5.3)

**Chapter 6: Personalizable Partial Automation** A personalizable form of partial automation for players with motor disabilities (Section 6.1.4), examples of partial automation in two rehabilitation games (Section 6.1.5), empirical evidence that partial automation can make rehabilitation games accessible to players with motor impairments (Section 6.2), and a thematic analysis of spinal cord injury rehabilitation patients' experiences of playing partially automated games (Section 6.3)

**Chapter 7: Automation Confusion** Examples of partial automation in two action games (Section 7.1.1), empirical evidence that partial automation can improve the accessibility of action games to non-gamers (Section 7.2), and a grounded theory of non-gamers' confusion when playing partially automated action games (Section 7.2.1)

**Chapter 8: Conclusion** A reference architecture for partial automation (Section 8.1) and an associated development toolkit demonstrated in a simple game (Section 8.2)

# **Part I**

# **Background**

# CHAPTER 2

---

## Motivation

---

**P**LAYING digital games is a popular pastime and cultural touchstone. Approximately 61% of Canadians play digital games [77] and approximately 74% of video game playing parents play with their children [78]. However, games can be difficult for some people to play. For example, *novices who are unfamiliar with games* and how they are controlled may have difficulty understanding what is happening in the game and deciding how to respond, while *players with motor disabilities* who are familiar with games may have difficulty controlling some games and be unable to play them. Although the reasons why gaming is difficult for these two types of players are fundamentally different, the same forms of player assistance may help them to play. Partial automation can control parts of games that players with motor disabilities cannot and can make gameplay decisions that

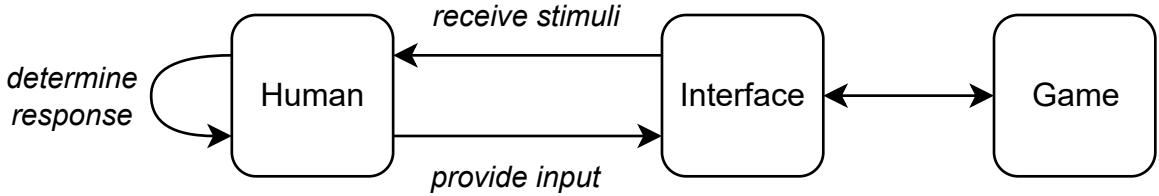


Figure 2.1: The game interaction model of Yuan et al. [329].

novices do not yet understand. This chapter identifies two groups of people who may benefit from partial automation, recounts the reasons why they play digital games, and advocates for improving games accessibility for these players.

## 2.1 Gameplay Barriers Facing Disabled Players

Users encounter barriers to access when interacting with a system demands abilities that users do not have [278]. For example, video games are inherently visual and players with vision impairments may be unable to see game elements rendered onscreen [302]. Games designed with the expectation that players have perfect vision may use text sizes that are too small for players with low vision to read or may use colors that are difficult for players with color blindness to distinguish [329]. Therefore, vision-impaired players may encounter barriers to access in some games.

When playing digital games, players first *receive stimuli* displayed onscreen, then *determine a response*, and finally *provide input* to the game using its controllers [329] (Figure 2.1). Providing input changes the game's state and consequently the stimuli players receive before once again determining a response and providing input. If a player is unable to perform even one of these steps, then they may be unable to play. Players with sensory disabilities, such as those caused by low vision [302], may be unable to receive stimuli displayed onscreen. Players

with cognitive disabilities, such as those caused by fetal alcohol spectrum disorder [268], may be unable to determine a response. Players with motor disabilities, such as those caused by spinal cord injury [52], may be unable to provide input by manipulating the game’s controller.

In this section, we will discuss the barriers to access encountered by two groups of disabled players: players with motor impairments and non-gamers. Although the term “non-gamer” is difficult to define [98; 39; 68], we define non-gamers as people who do not play games for their own enjoyment. We also view disability through the lens of social model of disability, so it may not yet be obvious why we consider non-gamers to be disabled. As we will explain in the rest of this chapter, non-gamers and players with motor impairments both have difficulty providing input to games and, as we will explain in Chapter 3, both can be assisted by partial automation. Non-gamers’ difficulty providing input is rooted in the complexities of determining responses in games: players have to use their knowledge of the game’s rules to decide what to do and their knowledge of the game’s control scheme to make that happen. Therefore, games that require players to quickly respond to in-game events and precisely manipulate a controller, which non-gamers are not yet proficient in using, are disabling for novices [304].

In the rest of this thesis, issues of accessibility will be discussed from the perspective of the social model of disability [185]. In contrast to the medical model, which considers the individual to be the locus of disability (i.e., people *are* disabled), the social model considers disability to be the result of barriers to access imposed by the individual’s social and built environment (i.e., people are disabled *by their environments*). In this way, non-gamers and players with motor impairments are both

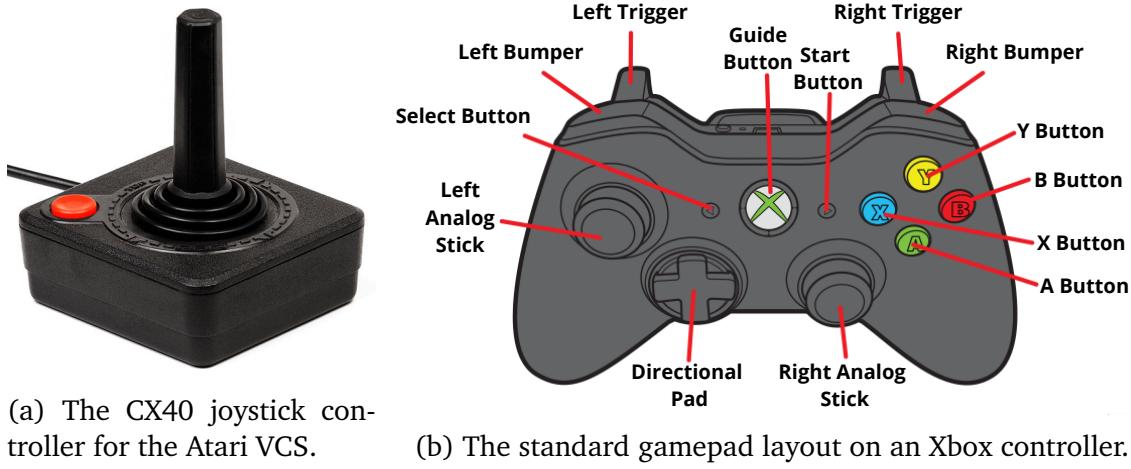


Figure 2.2: Example game controllers.

*disabled by games* that expect them to have abilities that they do not. While the medical model of disability might consider only persons with a medical condition that causes impairment to be person with a disability, we consider anyone who cannot conform to a game's interaction demands to be a disabled person.

### 2.1.1 Gameplay Barriers Facing Players with Motor Impairments

Although most recent games have been designed to be played using a mouse and keyboard or a gamepad, there is no universal interface that can be used to control all games. Each game places its own set of interactive demands on its players and, from the Atari VCS' simple joystick (Figure 2.2a) to the Xbox One's complex dual analog stick gamepad (Figure 2.2b), the interactive demands of games have increased over time [62]. This means that players with motor impairments enjoy only precarious access to games.



(a) The Wii Remote controller and Nunchuk expansion.



(b) *Kid playing Wii Bowling by daveynin* ([Flickr](#))

Figure 2.3: A child playing *Wii Sports* and the controller used to play it.

Dalgleish explains that as a child he played video games by placing his controller on the floor and manipulating its buttons with his right hand and left foot, because he did not have a left hand [62]. Since he was not required to hold the controller, the interactive demands of the games he played did not disable him. However, he could not employ this style of play to Wii games, which are controlled using two separate devices designed to be held in each hand (Figure 2.3a), and so was unable to play them. Dalgleish's experience of being disabled by Wii games illustrates how players who cannot use even one part of a game's controller may be summarily excluded from play. The interactive demands of controllers may be one of the most disabling aspects of games, since the majority of disabled gamers have motor impairments [246; 19].

Games involving physical activity, such *Wii Fit* [218] and *Wii Sports* [217] (Figure 2.4b), are designed to be physically demanding, which can limit their accessibility. For example, *Skyfarer* is a shoulder rehabilitation game that tasks players with performing external rotation, rowing, diagonal pull, and diagonal lift exercises [112]. Players who can perform some, but not all, of these exercises may be unable to play and benefit from these therapeutic movements. If, however, these games could be personalized somehow to remove the interaction demands that disable each individual player, then their benefits might be realized by players with a broader range of physical abilities. But, unfortunately, these sorts of modifications, such as a sleeve to help paraplegic players hold motion controllers [199] or a balance board that can be used by players with mobility aids [296], can require significant effort to engineer and may only extend access to a portion of the people who might benefit from the game. Overall, players with motor impairments are disabled by the inflexible interaction demands imposed by game controllers.

### 2.1.2 Gameplay Barriers Facing Non-gamers

Despite gaming's popularity, there are still many people who prefer not to play. These *non-gamers* have given many reasons why they do not play, citing a lack of interest in games, not finding them fun, and seeing them as a “waste of time” [68; 39]. While non-gamers are not attracted to games for their own enjoyment, they may have people in their lives, such as their friends and family, that they want to play games with [68]. However, games are designed to challenge players and for some these challenges can exclude [246; 62; 101].

Non-gamers have not spent the necessary time to learn gaming conventions and so are unfamiliar with the *language of games* [68]. Games are systems of rules [292], made of smaller games [170], and the fun of play is in learning the game’s underlying mathematical patterns [169; 316]. Non-gamers may not know, for example, that hearts represent their avatar’s health points or that red containers explode. They may have difficulty deciding what to do in the game and in what order. Since non-gamers may need to learn how to play a game while playing it, they may misinterpret what happens onscreen and come to learn the wrong lessons [250], making games even more difficult to understand.

Non-gamers also find game controllers difficult to use [68; 39; 98] and find complex input sequences especially difficult [105; 101]. The complexity of games and their controllers has increased dramatically over the last five decades [62]. For example, *Frogger* [165] can be played with a single joystick, whereas more recent games involve multiple sticks, triggers, buttons, and accelerometers. Non-gamers can find these newer devices daunting and may be unwilling to invest the hundreds of hours needed to learn them or may be unable to use them at all. In terms of the game interaction model shown in Figure 2.1, non-gamers face a range of cognitive barriers that make determining appropriate responses and providing input to games difficult. They are disabled by conventional representations of game elements (e.g., red explosives) and may have difficulty interpreting stimuli from the game. Non-gamers may not understand how to play games competently and may have difficulty deciding what to do, deciding what order to do it in, and doing so quickly enough to keep up with the pace of the game.

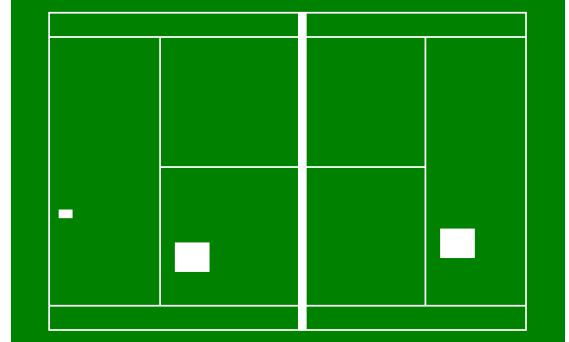
In summary, the interaction demands of games can disable players who do not have the abilities games require of them. Players with motor impairments may be unable to perform the bodily movements involved in play or use game controllers in the way they were designed to be used. Non-gamers have little or no gaming knowledge to rely on; they may not understand what they are supposed to do in the game or may have difficulty using a controller to make the game do what they want. Despite the sometimes significant barriers to access these types of players encounter, they may still find themselves compelled to play digital games in specific situations. In the next section, we discuss the reasons why non-gamers and players with motor impairments may want to play digital games.

## 2.2 The Uses of Digital Games

There are many reasons why disabled players may want to play digital games. Games are a powerful artistic medium; they can engender new experiences in players [56], enable them to engage with others in new ways [171], and motivate players to perform tasks that they might otherwise be uninterested in doing [258]. This makes games especially useful for both social and productive purposes. Many games are designed to be played cooperatively or competitively with multiple players [24]. Parents may use games as a way to connect with their children and groups of friends may play games as a way to foster togetherness [308]. Games also present challenges to players that push them to the limits of their abilities and games can motivate players to extend their abilities further. Some rehabilitation games—games designed for use in physical rehabilitation—have been shown to improve players' sensorimotor function and overall physical capacity [200; 201; 199].



(a) The Magnavox Odyssey. Its controller has three knobs; two for controlling the position of the player's dot and one for controlling the English on the ball.



(b) Tennis for the Magnavox Odyssey (emulated using [Odyemu](#))

Figure 2.4: The first video game console and one of its games.

These examples illustrate some of the positive gaming experiences that are missed by those excluded from play. In this section, we first describe how games are used by non-gamers and people with motor impairments as a social activity and then show how games can motivate patients to exercise in physical rehabilitation.

### 2.2.1 Gaming as a Social Activity

The first video game consoles (e.g., Home Pong [13] and Magnavox Odyssey [262]) made their home debut in the mid-to-late 1970's. They were simple devices, with only a few buttons and knobs [62] (Figure 2.4a), that came with a limited selection of preprogrammed sports-themed games designed for a young [66] and male [142] audience. As these consoles aged, so did their players. Today, digital games are far more complex and the contexts in which they are played are far more varied than they were in the early days. Nintendo's family-oriented Wii home console, released in 2006, ushered in a new era of video game design that once again invited the Atari generation, who were by then in their 40's and 50's [66], to play with

their children and grandchildren. Games such as *Wii Sports* [217], designed for players of different ages and levels of physical ability [47], gained popularity and video game consoles became the *digital hearth* of the household [309]. This section explores gaming as a social activity and identifies the reasons why people choose to play together. Although both non-gamers and players with motor impairments play games as a social activity, this section is concerned primarily with non-gamers.

### 2.2.1.1 The Social Motives of Play

Unlike other social activities involving entertainment media, such as seeing a play or film, digital gaming is inherently interactive and affords opportunities for players to interact with each other in novel and delightful ways. While the reasons why players choose to play with others may be different, all players may use games as a resource that they leverage for their own social purposes [1]. For gamers—people who choose to play games for their own enjoyment—games provide an “excuse” [308] to socialize with others. In 2020, during the COVID-19 pandemic, approximately 43% of Canadian adult gamers and 70% of teen gamers used digital games as a way to stay socially connected to friends and family while physically isolated [78]. Playing together has been shown to enhance both players’ enjoyment of games [96; 95; 153] and the quality of players’ relationships [230]. Social play at large-scale events in particular, such as tournaments and LAN parties, can be socially validating for dedicated gamers and provides a forum for the exchange of gaming knowledge [144; 4]. It is perhaps unsurprising that people who enjoy playing digital games want to play games with others who share their interests; however, it is not only gamers who choose to play for social purposes.

With the majority of Canadians regularly playing digital games [77], those who choose not to play—so called *non-gamers*—may be compelled to partake in gaming activities at the behest of their gamer friends and family. One often considered scenario involves families comprised of members from different generations and with different levels of gaming expertise, playing together to promote family bonding [47; 313]. Parents and grandparents have reportedly used games as a means to connect with their gamer children and “*bridge the gap*” between their interests [308]. Parents may also choose to play games with their children to monitor and control their exposure to potentially harmful content in games [47; 276; 173]. Conversely, children reportedly use games as a means to invert intergenerational hierarchies [1; 49], instructing their elders and performing the role of mentor, since children typically have greater expertise in the domain of gaming [308; 309]. Due to the many potential differences between gamers and non-gamers, as well as within intergenerational families, games for social play must be designed with consideration for differences in players’ interests [144; 171], physical abilities [156; 157; 194; 155], and familiarity with gaming technologies [256; 251].

Non-gamers play digital games because they want to play with their gamer friends and family. Every so often a game designed with consideration for a casual non-gamer audience, such as *Wii Sports* [217], becomes popular among gamers and both groups can play together. But it may be that gamers will most often invite their non-gamer friends and family to play games that were not designed for them. If we want to improve the accessibility of digital games to non-gamers and players with motor impairments, then we need to find ways to broaden the accessibility of popular commercial games to players who may have greater difficulty using some

parts of the game's controller, understanding some of the tasks involved in play, and making some decisions regarding how to perform these tasks. In the Chapter 3, we will explain how partial automation might alleviate some of these difficulties.

### 2.2.2 Gaming in Physical Rehabilitation

As a second example to motivate improving the accessibility of digital games, in this section we describe how games can contribute to the rehabilitation therapy of people with motor impairments, such as patients in spinal cord injury rehabilitation. Although games are played primarily because they are fun [78] some games have been designed for more serious purposes, such as physical rehabilitation [258]. *Exergames* are digital games that are controlled by performing exercises [210]. They are closely related to *active games*, which are games involving movement [257], and *exertion games*, which are games involving physical effort [210]. While exergames, such as *Wii Fit* [218], are used by a small percentage of gamers [78] as a way to motivate themselves to exercise, there are patients in rehabilitation services for whom exercise is not a choice, whose experiences of physical rehabilitation might be improved by exergaming.

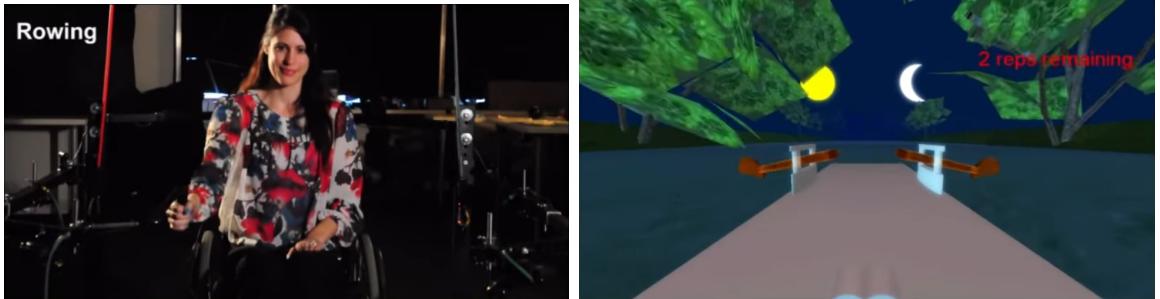
In this section, we will describe the physical rehabilitation of one specific group of people with motor impairments: people with *spinal cord injury*. We chose to examine this group because it is especially difficult to design exergames that are accessible to them. People with spinal cord injury can have radically different physical abilities; for some their motor function is considered normal while others may have no sensation or motor function at all below their necks [254]. As we will explain in Chapter 3, the extreme variability in the motor abilities of players with spinal cord

injury greatly constrains the design of games that are accessible to this population and precludes the design of exergames, which are physically demanding by definition. We will now summarize the use of games to improve patients' rehabilitation outcomes and experiences. Although both non-gamers and players with motor impairments may undergo physical rehabilitation, this section is concerned primarily with players with motor impairments.

### 2.2.2.1 Spinal Cord Injury Rehabilitation Gaming

Spinal cord injury rehabilitation seeks to improve patients' physical capacity and functional performance so that they can live healthy lives with independence [128; 140; 162; 215]. However, this requires patients to perform exercises that are often considered “*boring*” and “*monotonous*.<sup>1</sup>” [200; 201; 202; 224; 23; 22] Rehabilitation games can improve patients' experiences of exercising by incorporating rehabilitation exercises into players' control of the games. Here, we present three classes of rehabilitation games designed to help players improve their functional performance and physical capacity.

**Balancing and Reaching Games:** Spinal cord injury can cause motor impairment in the hips and trunk [161; 254; 140], which can make balancing and reaching tasks more difficult [22; 23]. Balancing and reaching rehabilitation games turn these, possibly uncomfortable [23], movements into game controls. For example, *Wii Fit* [195; 196; 311] and similar standing [22] and sitting [23] balance games use pressure sensors that players activate by leaning to control the games. Reaching games use camera-based motion sensors to play games that involve reaching out and touching virtual objects superimposed



(a) A player performing rowing exercises while playing *Skyfarer*. (b) A screenshot of rowing in *Skyfarer*. Rowing makes the player's boat move forward.

Figure 2.5: The *Skyfarer* shoulder mobility game.

over video of the player [162; 85]. These games provide players fun and compelling ways to train functional motor skills, thereby improving players' quality of life.

**Shoulder Mobility Games:** *Skyfarer* (Figure 2.5) is a mixed reality game that integrates exercises from the STOMPS protocol, a shoulder exercise program for manual wheelchair users [212], into the player's control [113; 111; 112]. Players navigate the game world in a seafaring vessel [112] by performing rowing, external rotation, diagonal pull-down, and vertical lift exercises [111] using resistance bands [113]. The shoulder exercises that players perform control in-game activities that mirror the player's physical movements. For example, players row to move their vessel forward and do external rotation movements to pull buckets of water out of the sea [112]. Since the creation of *Skyfarer*, another game called *Dash Lane* has been designed around shoulder exercises recommended for wheelchair users [198; 197]. Players drive along multi-lane tracks while shooting and switching lanes to avoid obstacles. Players raise their arms to shoot and punch to the left or right to switch lanes.

Games such as *Skyfarer* and *Dash Lane* provide players a more enjoyable and compelling experience of performing shoulder exercises by transforming the player’s physical activity into movement-based metaphors for gameplay activities [112].

**Wheelchair & Arm Ergometry Games:** GAME<sup>Wheels</sup> and GAME<sup>Cycle</sup> enable players to control digital games using their manual wheelchair or an arm ergometer [305; 224; 225; 84; 116; 83]. Players using GAME<sup>Wheels</sup> push their wheelchairs on metal rollers that keep them in place. The device uses the rotation speed of each wheel to drive movement in games such as *Need for Speed 2* [84]. GAME<sup>Cycle</sup> is used by cranking an ergometer’s hand pedals and tilting the ergometer from side to side to turn. Playing games using these interfaces has been shown to produce physiological responses that are similar to or better than responses elicited by wheelchair or arm ergometry alone [83; 226; 224].

Other active games have proven effective in spinal cord injury rehabilitation. Active games have been shown to improve the physical capacity [258; 41; 255; 200; 199; 145; 94] and functional performance [311; 195; 196; 85; 162] of players with spinal cord injury. These findings led Mat Rosly et al. to conclude that active gaming can be considered to be at least as good as traditional exercise in a clinical setting [199]. However, the accessibility of these games is a major factor limiting their application. In Chapter 6, we will present a form of partial automation that can make rehabilitation games more accessible to players who cannot perform some of the game’s exercises.

## 2.3 Summary

Games have important applications beyond providing a fun leisure activity. For friends and families who play as a social activity, games provide novel ways to interact with each other that promote bonding. For patients in rehabilitation, who play as a physical activity, games provide a challenging and compelling means to improve players' experiences of performing routine exercises. When players are able to access games for these important purposes, games can improve and enrich players' lives.

Barriers to access faced by non-gamers and players with motor disabilities are a major factor limiting the use of games for social and medical purposes. Designers can take several approaches to improve the accessibility of their games to these types of players but, as we will discuss in the next chapter, each has limits. Ultimately, all games place interaction demands on their players and these demands are only flexible up to a point. From shooting in *Space Invaders* [289] to throwing a ball in *Wii Sports* [217], there is always something that players need to be able to do to play. In the next chapter, we will describe several approaches that designers can take to improve and broaden the accessibility of their games.

# CHAPTER 3

---

## Games Accessibility & Partial Automation

---

**G**AMES accessibility is concerned with identifying, reducing, and eliminating barriers to access in games. In the previous chapter, we explained that digital games present barriers to access to players who do not have the abilities designers expect them to have. Games that require players to make complicated gameplay decisions or perform demanding bodily movements disable players who cannot do these things. We also explained that non-gamers are unfamiliar with games and may encounter barriers related to the complexities of making gameplay decisions and providing input to the game. Games are usually designed for people who like games; they, therefore, set a high barrier to entry for non-gamers. Lastly, we explained that players with motor impairments may encounter barriers to access in games related to their abilities to use the game's controller or

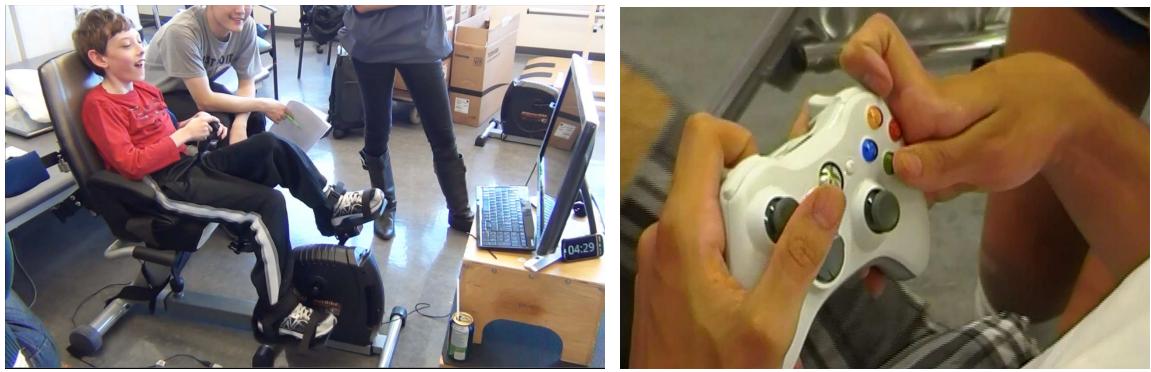
perform the bodily movements involved in play. Games are often designed for people with supposedly normal motor abilities and may disable players whose motor function deviates from what designers consider normal. In this chapter, we will describe prior approaches to accessible game design, introduce *partial automation*—the games accessibility approach explored in the rest of this thesis—and explain how these approaches can be combined to extend the accessibility they afford.

## 3.1 Approaches to Designing Accessible Games

Designers can employ the following approaches to improve the accessibility of their games. Designers can adhere to the principles of *universal design*, and design for the abilities held by the entire target audience. Games can also be made more accessible through *interface adaptation*, either through alternative interfaces designed for players with similar abilities, or through personalization of interfaces to the abilities of individual players. Additionally, players for whom controlling the game is too difficult can be assisted using *player balancing* techniques that adjust the difficulty of the game’s challenges. In this section, we describe these three approaches to improving a game’s accessibility.

### 3.1.1 Universal Design

Universal design advocates that a product should be accessible to everyone without adaptation [278]. For example, the *Liberi* exergames [127; 126] (Figure 3.1a) were designed for children with cerebral palsy who can pedal a bicycle, tilt an analog stick, and press a button (Figure 3.1b). This means that *Liberi* is accessible

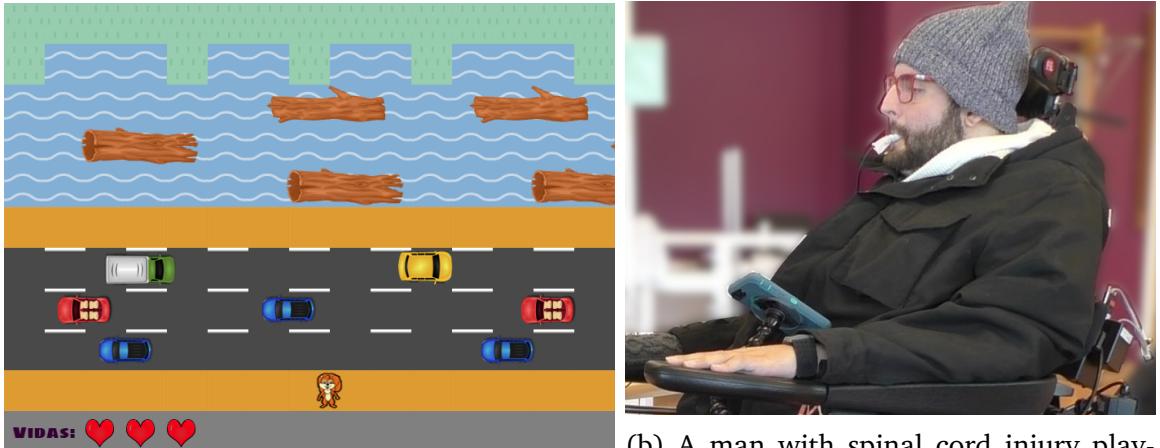


(a) A child with cerebral palsy playing *Liberi* using a gamepad controller and a pedalling device [127] (used with permission). (b) A child with cerebral palsy holding a gamepad controller [126] (used with permission).

Figure 3.1: Children playing *Liberi*.

to everyone for whom it was designed; that is, most children with cerebral palsy. Games for non-gamers, such as those described in Chapter 2, are often designed from a universal design perspective, as they need to be accessible to players with a range of physical abilities. Games designed through universal design present a one-size-fits-all interface, but because no one size can truly fit everyone [312; 324; 323], players outside the target group may be excluded [62]. For example, children with cerebral palsy who cannot pedal or use a gamepad controller cannot play *Liberi* on their own. Designers can further restrict the abilities necessary to play, broadening the game’s accessibility, but the resulting game may be seen as too simple for players who can do more.

There are two approaches to the universal design of games: *broad accessibility*, requiring only abilities that are widely held (e.g., press a button) and *focused accessibility*, requiring only specific abilities held by target players (e.g., use of a manual



(a) A screenshot from the *Frogger*-like *Zac - O Esquilo*. (b) A man with spinal cord injury playing *Liberi* using a bite switch—a mouth-controlled button.

Figure 3.2: One-switch play.

wheelchair). As we will now discuss, both approaches make trade-offs to achieve universal accessibility.

### 3.1.1.1 Broad Accessibility

Games played using a single button or switch serve as exemplars of how broad accessibility can be achieved through universal design [330]. These one-switch games reduce interaction to the bare minimum of clicking a single button. Although one-switch games are nearly universally accessible, the design of these games is constrained by the simplicity of their common interface. For example, *Zac - O Esquilo* [203] (Figure 3.2a) is a one-switch version of the arcade game *Frogger* [165]. When the player presses the button, an algorithm chooses in which direction to move the avatar. Although this type of transformation through automation can make a game more accessible, a player who is capable of controlling a joystick



(a) A man with spinal cord injury playing *Liberi* in his manual wheelchair. (b) A woman with spinal cord injury playing *Liberi* in her power wheelchair.

Figure 3.3: People with different types of mobility aids playing *Liberi*.

might prefer the original version where it is possible to control the character's direction of movement. Thus universal accessibility comes at a cost to players with richer physical abilities.

Another class of broadly accessible games are games that players control with bodily movement. These ask players to run, walk, or push their wheelchairs around a physical space and use this movement to control the game. These games advantage players with superior mobility but remain accessible to anyone who can move around. For example, in *iGYM*, players move around the field of play to bounce a disc of light, projected onto the floor from above, into their opponent's net [114]. Play is simple, like *Air Hockey*, and that simplicity enables players who use walkers, wheelchairs (Figure 3.3), or no assistive device at all to play together. The same can be said of *Powered to Play*, a GPS-enabled, mixed-reality, capture the flag game designed for players who use powered mobility devices [73]. These games are accessible to players with vastly different abilities because only the ability to move

around is required. However, players with limited mobility may need assistance to compete with more mobile opponents.

### 3.1.1.2 Focused Accessibility

Designers may choose to improve the accessibility of their game by narrowing its target population to players with specific abilities, such as the ability to pedal a bicycle or use a wheelchair. For example, the *Liberi* exergames were designed for children with cerebral palsy *who can pedal a stationary bicycle and use a gamepad controller* [127; 126]. This is different from designing for players with specific *conditions* that may cause disability, such as cerebral palsy or fetal alcohol spectrum disorder, whose abilities are unknown and in need of discovery [268]. For this approach to work, games need to be designed with consideration for what players are able to do, not what they are unable to do. Although these games are designed to overcome accessibility barriers encountered by disabled players, they extend access to a more focused sub-population defined in terms of their abilities. This is why audio games (see Figure 3.4a) typically target players with vision impairments but are actually designed for players who can hear [302]. In this way, focused accessibility enables designers to make games accessible to a specific group of players with disabilities by designing for their abilities.

Many games have been designed specifically for players who use wheelchairs. Typically, these games use specialized hardware to turn the player's wheelchair into a game controller. For example, as described in Chapter 2, GAME<sup>Wheels</sup> uses wheelchair propulsion to control computer games [84; 305]. Players secure their wheelchair to a platform with rollers that control joystick interaction with the game.



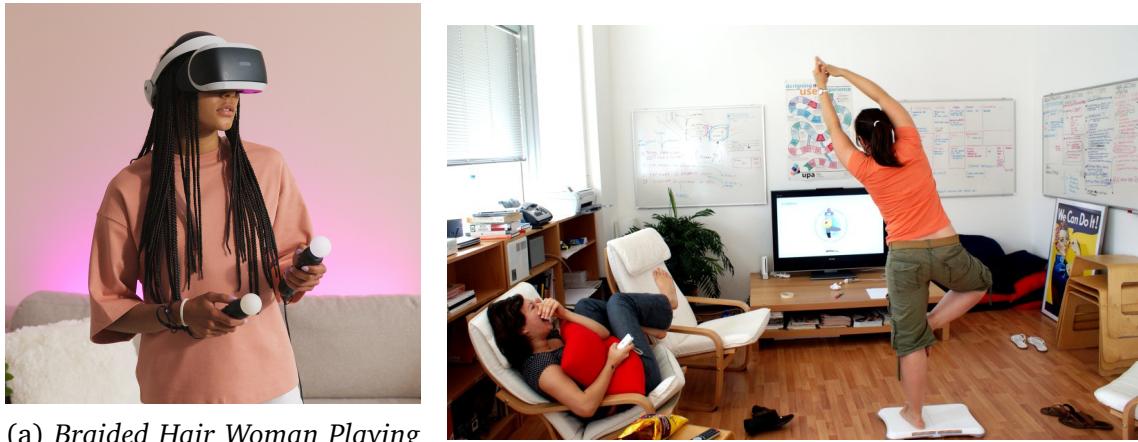
(a) A screenshot from the audio game *Audio Pong* [247]. This game produces no visual stimuli.

(b) *Kinect* by Wipley/BrainSins (Flickr). The Kinect camera-based sensor sits just below the television.

Figure 3.4: Games with focused accessibility.

Manual wheelchair gaming interfaces and games have also been created using the Kinect [102; 103; 129; 100] (Figure 3.4b) and wheelchair mounted accelerometers [61]. These games are of course limited to players who can control a manual or power wheelchair.

Other focused games, described in Chapter 2, have been designed for players engaging in common rehabilitation exercises. For example, *Skyfarer* [113; 111; 112] is a mixed reality game that incorporates exercises from the STOMPS shoulder exercise protocol [212] into its control scheme. The player performs rowing exercises to navigate a sea-faring vessel, and collects water in a bucket by performing an external rotation exercise. Other games for rehabilitation require players to perform upper body exercises for wheelchair users [197], static balancing [22], seated balancing [23], calf raises [267], and arm ergometry [116]. These games are designed for persons requiring specific rehabilitation exercises, and so are inherently limited to those who can perform those exercises.



(a) *Braided Hair Woman Playing PlayStation Vr* by Artem Podrez ([Pexels](#)). One Playstation Move controller is held in each hand.

(b) *Pausa pranzo senza il capo!!* by enfad ([Flickr](#)). The player is playing *Wii Fit* and standing on the Balance Board.

Figure 3.5: People playing active games.

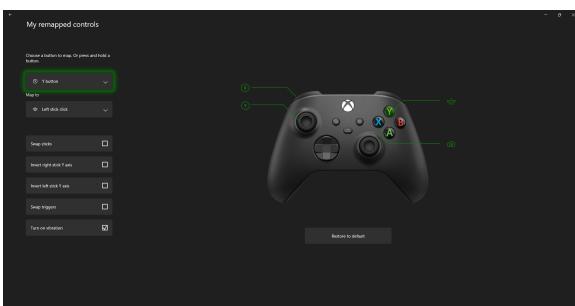
In summary, universal design is successful in creating inclusive games that a wide set of people can play. However, broadly accessible games may still advantage some players over others and games with focused accessibility exclude players whose abilities are different than the target group. As we shall see in the next section, adaptations to a game’s interface can help to overcome both of these limitations.

### 3.1.2 Interface Adaptation

Players who are disabled by a game’s controls are unable to play without an accessible alternative. Interface adaptation provides an alternative hardware interface designed around the abilities of excluded players. When many players are unable to use the same part of a game’s controller, designers can create an *alternative interface* to overcome the barriers that these players encounter. When making rehabilitation

games more accessible, designers need to enable players to perform the same exercises using the new interface. For example, Mat Rosly et al. designed a paddle-like sleeve for the PlayStation Move (Figure 3.5a) that makes it easier for players with tetraplegia to press buttons in a kayaking game [199]. Thirumalai et al. adapted the *Wii Fit* [218] Balance Board (Figure 3.5b) for players who use mobility devices such as walkers and wheelchairs [296]. The adapted Balance Board eases play for persons with mobility disabilities through a ramp for wheelchair access, a large platform, handrails to aid balance, a dedicated “jump” button, and adjustable center of balance sensors [195; 196]. This device made *Wii Fit* accessible to players with a broader range of motor abilities, from those who could stand and balance using a handrail to those who could lean in their wheelchairs. This extends the group that can play, but the adapted Balance Board may still be inaccessible to players whose abilities are different from the target population. For example, a player with complete tetraplegia may have insufficient center of balance control to overcome balancing challenges. Although creating alternative interfaces can greatly improve game accessibility for players with homogeneous abilities, designers may need to make multiple interfaces, incurring additional development overhead, to provide broader accessibility. This limits the efficacy of this approach for adapting existing games for populations with large differences in individual ability.

When players are disabled by different aspects of the game’s interface, it may be possible for individuals to play using a *personalized interface* made up of multiple accessible controllers. Heuristics for accessible game design, such as the Game Accessibility Guidelines [97] and Accessible Player Experiences [295], promote features allowing players to remap inaccessible game inputs (e.g., making an avatar



(a) The control remapping menu from the Xbox Accessories app.



(b) An adapted interface connected to an Xbox adaptive controller. (credit: [All Access Life](#))

Figure 3.6: Adapted interfaces.

move around or shoot) to accessible parts of a game’s controller (Figure 3.6a). This can greatly increase games accessibility for players who can use the default controller with some difficulty, but is incapable of making games accessible to players who cannot use the controller at all. Solutions such as the Xbox Adaptive Controller (Figure 3.6b), the interface device of Iacopetti et al. [136], and the AsTeRICS framework [231] overcome this limitation by enabling players to create their own bespoke interfaces. They allow players to control each of a game’s inputs using a separate, accessible device, enabling players with different abilities to play the same games by virtue of personalization. However, this approach requires that players control all aspects of play, which may not be possible if accessible alternatives are unavailable or are themselves too difficult to use.

### 3.1.3 Player Balancing

Another major form of game adaptation is player balancing. Player balancing techniques compensate for deficits in players’ performance, making games easier to

play and enabling weaker players to compete with stronger opponents [17]. Unlike game balancing, which presents challenges of the same difficulty to all players, player balancing personalizes a game's difficulty to the abilities of individuals. For example, first-person shooter games have historically provided *aim assistance* for players who aim using analog sticks, since analog sticks are less precise than the mice used by other competitors [306]. As with interface adaptation, designers using player balancing to make serious games more accessible need to ensure that aspects of play that are essential to a game's serious purpose (e.g. the exercise in an exergame) are still challenging enough to provide benefit.

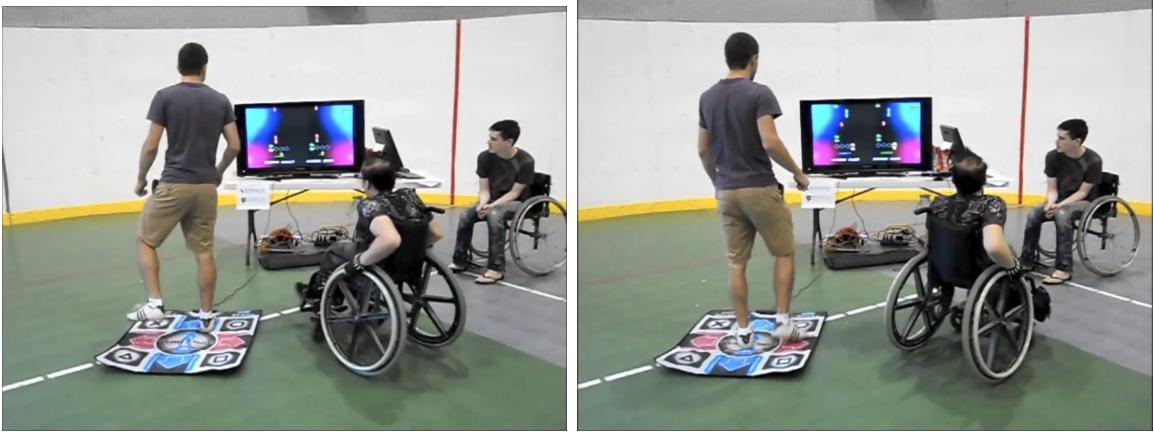
Balancing for player skill has been shown to enable more engaging social play [134; 211], increase relatedness among competitors [146; 104], and reduce differences in performance [17; 134; 146]. In a study by Hwang et al., participants with different levels of fine-motor and gross-motor function competed in a cycling-based racing game and an analog-stick-based shooting game, either with or without balancing for differences in players' motor abilities [134]. It was found that competitions in which balancing was used had closer outcomes and were perceived as more fair by players. These results indicate that player balancing techniques can be used to help players to overcome challenges that would be too difficult otherwise. However, player balancing is inherently limited to situations where players have at least some ability to use the provided interface. For example, Hwang's balancing techniques do not work for players who cannot cycle or use an analog stick at all.

In summary, universal design, interface adaptation, and player balancing can broaden a game's accessibility to disabled players. However, all of these approaches

still require players to control all aspects of the game, albeit in a more accessible way. In reality, players may not have the supposedly universal abilities designers expect them to or there may be no accessible alternative interface that lets the player fully control the game; and players cannot be assisted in controlling the game if they cannot control it at all. These approaches can extend a game's accessibility and when combined they can overcome these limitations and extend the game's accessibility even further. In the next section, we describe how even greater accessibility can be achieved through combining approaches.

## 3.2 Combining Approaches

Interface adaptation and player balancing can improve the accessibility of games, but both have limitations and each addresses different types of barriers to access. When combined, however, each technique can complement and help to overcome the limitations of the other. This was the approach taken by Gerling et al. when designing *Wheelchair Revolution* [104], a clone of *Dance Dance Revolution* [20] adapted for players who use wheelchairs. In *Dance Dance Revolution*, players stomp in time with music on an array of buttons, called a dance mat. To enable play using a wheelchair, a Kinect-based action recognition system, called KINECT<sup>Wheels</sup> [102], was used to transform wheelchair movements into game inputs. This alternative interface enabled players who used wheelchairs to play the game with players who used the dance mat, but it was unknown whether competition between players with such radically different motor abilities would be fair. To account for their differences, Gerling et al. used player balancing algorithms that decreased the number of movements required of weaker players, made timing movements easier, and



(a) Hitting a left arrow.

(b) Hitting a right arrow.

Figure 3.7: People playing *Wheelchair Revolution* [102] using the default dance mat controller (left) and the alternative KINECT<sup>Wheels</sup> controller (right).

scaled scores by a personalized score multiplier. Although dance mat players generally performed better than their wheelchair player opponents, player balancing had a positive effect both on competitors' feelings about competing against opponents with different abilities and on wheelchair players' experiences of enjoyment, autonomy, competence, and relatedness during play.

These results indicate that games that employ interface adaptation and player balancing can be accessible to players with different motor abilities. When players are able to play the game in their own way, differences in physical ability have less of an effect on players' performance and experience. KINECT<sup>Wheels</sup> broadens *Wheelchair Revolution*'s accessibility to players who can use a manual wheelchair, but not a dance mat. Player balancing algorithms overcome differences in players' performance caused by differences in the interfaces they use to provide a more balanced competition. Without KINECT<sup>Wheels</sup>, players' performance could not be balanced. Without player balancing, competitions between dance mat users and

wheelchair users may not be fair. Although *Wheelchair Revolution* is demonstrably more accessible than *Dance Dance Revolution*, it still does not provide truly universal accessibility. Some players may be unable to use either interface (e.g., power wheelchair users); some may be unable to do all of the game’s movements (e.g., tetraplegic players), and some may only be able to do these movements intermittently or for short periods of time (e.g., players with repetitive strain injuries). In the next section, we introduce *partial automation*—a novel approach to improving games accessibility that uses an AI copilot to remove and reduce interaction demands that present barriers to access for individual players.

### 3.3 Partial Automation

Due to the limitations inherent in existing approaches to accessible game design, additional games accessibility techniques are needed. Players for whom no adapted interface is accessible are unable to play and therefore cannot benefit from player balancing. For some players, no interface adaptation or player balancing technique can make all aspects of a game accessible. This is the problem addressed in this thesis: *how can games be made accessible to players who are unable to provide input?*

We have coined the term *partial automation* to describe an approach to games accessibility that simplifies games by automating control of part of their interfaces. The core idea of partial automation is simple: a player controls those parts of a game’s interface that they can while a *copilot*—an artificially intelligent agent—controls those parts of the interface that the player cannot control or has difficulty controlling. The game provides an interface controlled via some set of input devices, such as buttons or analog sticks. These inputs correspond to *game inputs*

that control different mechanics, such as setting the avatar’s speed of movement, direction of movement, or performing actions such as jumping or shooting. If, for example, direction of movement of a player’s avatar is specified using an analog stick, players who can use an analog stick provide that input themselves. For players who cannot use an analog stick, the copilot instead provides movement inputs to the game. This allows players to engage in games even when they cannot control all aspects of play, and to personalize the interface to whatever hardware they can use. Players must be able to control at least one game input, as otherwise gameplay would be completely passive. In this way, partial automation relaxes the interaction demands of games by reducing the number of controls that players have to learn and the complexity of players’ decision-making. Unlike other approaches to accessible game design, partial automation’s capacity to improve the accessibility of digital games has not yet been studied academically, although accessibility features similar to partial automation can be found in a small number of commercial games (e.g., *Bayonetta* [245], *Mario Kart 8 Deluxe* [219], *Kingdom Come: Deliverance* [315], *Zone of the Enders: The 2nd Runner - M&RS* [166], *Street Fighter 6* [43]).

The assistance provided by partial automation is especially potent in *action* games—real-time games in which the player makes an avatar perform actions to overcome challenges. For example, the game *Bayonetta* [245] provides an “Automatic” mode that automates attacking and avatar movement. When playing normally, the player is tasked with fighting groups of enemies by tilting an analog stick to move the avatar and pressing buttons to make the avatar do different attacks. When played with in “Automatic” mode, the player is tasked with pressing one button, to make the avatar do an attack, while partial automation selects which attack

to do and which enemy to attack. This reduces players' control to a single button and simplifies players' decision making, since they only need to decide when to do an attack.

Under partial automation, players share control of their avatar with an AI partner. Any AI algorithm, or collection of algorithms, can be used for this purpose, so long as the copilot can control any combination of the game's inputs. For simple games, copilots' plans could be expressed using planning models [205], such as hierarchical finite state machines (HFSM), behaviour trees, or goal-oriented action planning (GOAP). In more complicated games, copilots' policies could be learned [328] as a belief-desire-intention model [79], a deterministic policy through deep Q-learning [208], or a stochastic policy through proximal policy optimization [269]. As a model-agnostic approach, partial automation affords designers limitless freedom to choose AI algorithms that cooperate well with players.

There are two forms of partial automation: *input automation* and *one-switch* [52] (Figure 3.8). Input automation assists players by taking over some of the game's inputs. For example, Hwang et al. balanced a shooting game for differences in players' manual abilities by aiming directly towards the most likely target when the player shoots [134]. Cechanowicz et al. helped players align their steering with the road in a racing game [46]. Hougaard et al. helped players more reliably control an infinite runner game using a brain computer interface by making the avatar jump on its own [130]. An example of input automation is shown in Figure 3.8; the player controls one of the game's inputs while partial automation controls the other. In contrast, one-switch games assist players by reducing their control to a single button [74; 329]. For example, *Zac - O Esquilo* is a one-switch adaptation

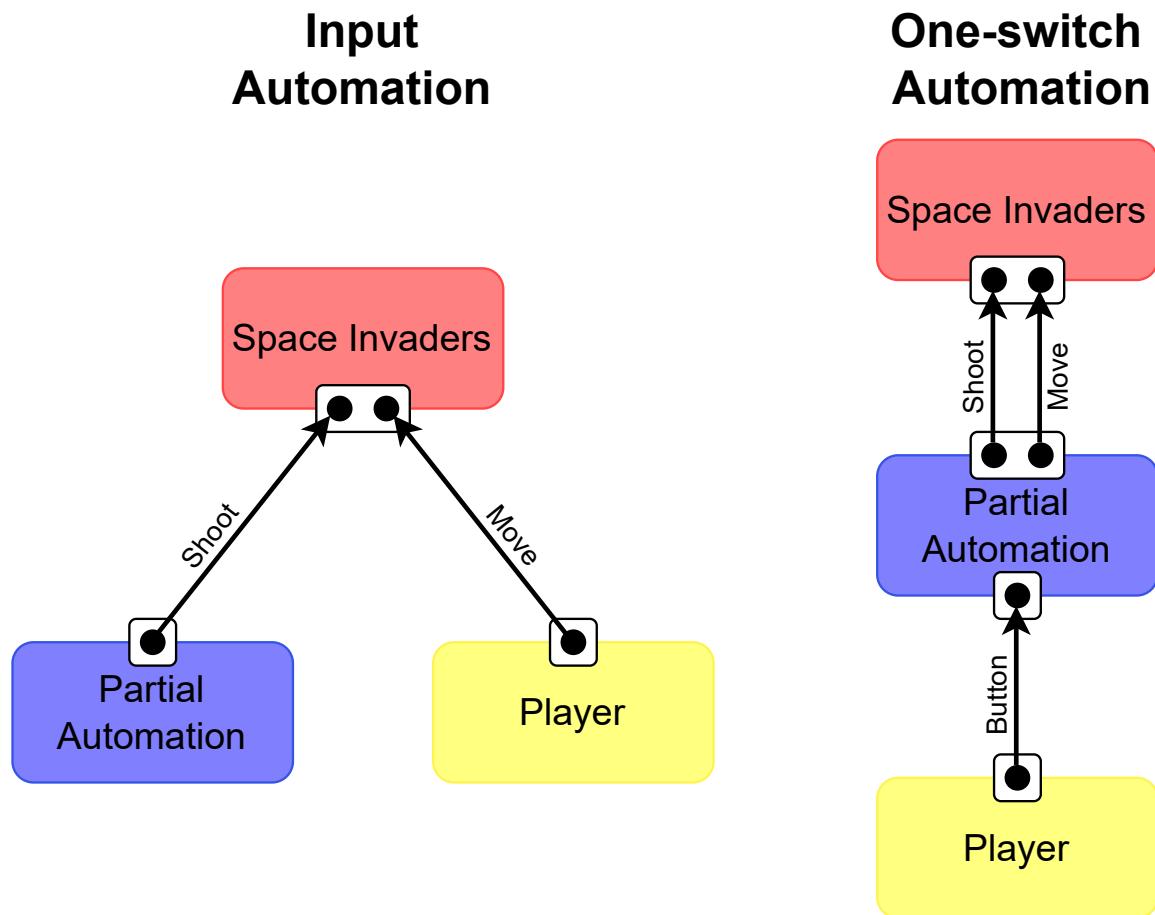


Figure 3.8: Two examples of input automation (left) and one-switch automation (right) in *Space Invaders*. The game inputs in *Space Invaders* are called *shoot*, which makes the avatar shoot, and *move*, which makes the avatar move left and right.

of *Frogger* [165] in which an algorithm chooses which direction the avatar moves when the player presses the button [203]. This form of partial automation is radically different than input automation. As depicted in Figure 3.8, one-switch partial automation uses input that the player can provide to guide the copilot’s control of the game’s inputs.

Partial automation is related to, but distinct from, several other forms of player assistance. As discussed in Section 3.1.3, *Player balancing* helps weaker players compete with stronger players [17] and some player balancing solutions use partial automation (e.g., [134; 46]) while others do not (e.g., [104]). *Dynamic difficulty adjustment* changes game mechanics in response to individual players’ abilities [133], buffing players or spawning fewer enemies but not automating control. And finally, *full automation* automates control of all player tasks, so that both player and automation control the game at different times. For example, the “Auto-Battle” feature in *Fire Emblem: Three Houses* [138] fully automates battles, selecting actions that adhere to the player’s chosen strategy.

To summarize, partial automation is a promising approach to games accessibility that may improve the accessibility of games for non-gamers, who have difficulty using controllers and making gameplay decisions, as well as players with motor impairments, who may be unable to use a game’s controllers. Although partial automation has previously been used in a handful of commercial games, we are the first to study this potentially game changing accessibility technique. In Chapter 6, we will present a novel form of partial automation that enables players with motor impairments to personalize both the physical interface they use to play, through

interface adaptation, and their control of the game’s inputs, through input automation. We have found that combining partial automation with other techniques can extend a game’s accessibility further than either can on its own. In the next section, we present a novel ontology of approaches to games accessibility and explain how partial automation can extend the accessibility other approaches.

### 3.4 Ontology of Approaches to Games Accessibility

Figure 3.9 depicts the approaches to games accessibility described in this chapter as well as relations between them. We say that interface adaptation and player balancing can *extend* the accessibility afforded by universal design, because they may overcome universal design’s limitations. As explained in Section 3.1.1, universal design produces games that require only the abilities that are held by all target players. But players may not be able to do what designers of universally designed games expect them to and interface adaptation may enable players to do the same things using different interfaces. Even when all players have the abilities that designers expect them to, players may perform at different skill levels and player balancing may help players of lower skill to compete with players of higher skill.

As explained in Section 3.2, interface adaptation and player balancing can *complement* each other by extending the accessibility of each approach using the other. Partial automation can in turn extend the accessibility afforded by interface adaptation and player balancing even further. Players who can control only some of a game’s inputs using an alternative interface may be unable to play at all unless control of the inputs they cannot control is delegated to partial automation. Likewise, player balancing cannot assist players who are unable to control the game. Partial

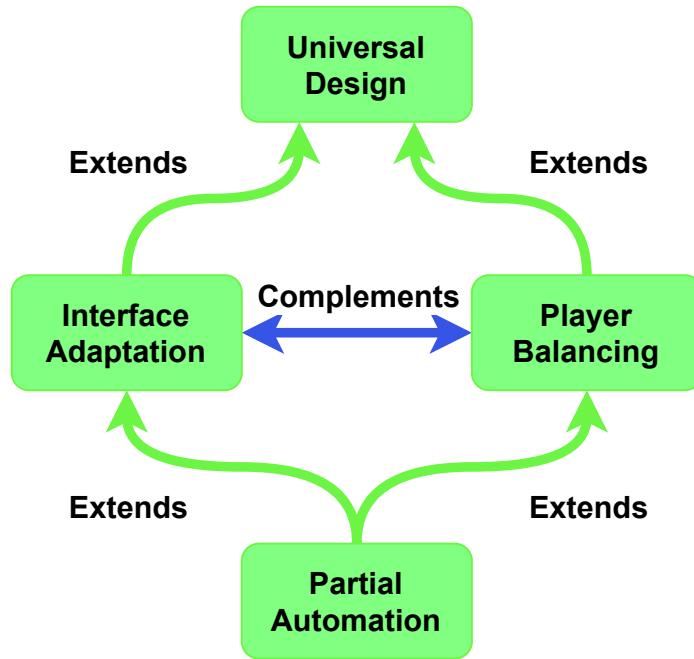


Figure 3.9: A novel ontology of approaches to games accessibility and relations between them. *Extends* means that one approach overcomes the limitations of another and *complements* means that both approaches overcome each other's limitations. For example, as explained in Section 3.2, each of interface adaptation and player balancing *overcome the limitations of* the other, but neither interface adaptation nor player balancing *overcome the limitations of* partial automation.

automation can extend the assistance afforded by player balancing to players who could not be assisted otherwise. We will demonstrate partial automation's capacity to extend the accessibility afforded by interface adaptation and revisit this ontology in Chapter 6.

### 3.5 Summary

There are several approaches that designers can take to improve the accessibility of their games, but each has limits. Universal design excludes players without the

requisite abilities; player balancing can assist but not replace, and interface adaptation may necessitate the design of bespoke interfaces for each player in the worst case. Partial automation uses an AI copilot to automate control of game inputs that players are unable to control, which can further extend the accessibility afforded by these other approaches. In his book about one-switch games, Ellis writes that “*In accessible gaming, this method of sharing controls between a team is invaluable. If at least one button does something fun or important, then there is a way.*” [74] In the next chapter, we discuss the design of interactive systems, such as partially automated games, in which human users share control with AI agents.

# CHAPTER 4

---

## Human-AI Shared Control & Human-Machine Cooperation

---

In Chapter 6, we will investigate how partial automation may improve the accessibility of digital games to non-gamers and players with motor impairments. Partial automation makes decisions for players and helps them to control game inputs that they find difficult to use. An AI copilot might, as in *Bayonetta* [245] for example, select which enemy the player should attack and use the game input that controls the camera to place that enemy in the centre of the screen. With partial automation, the player and the copilot cooperatively share control of the game. This makes partial automation a form of human-AI shared control—an emerging interaction paradigm in which systems are controlled through the tightly-coupled cooperation of a human and an AI agent [2; 288]. In other application

domains, such as semi-autonomous driving and computer assisted surgery, sharing control with AI can help users to perform their tasks more safely, precisely, and reliably. As we will explain in this chapter, designers of partial automation can take inspiration from uses of human-AI shared control in other domains and draw insights from human-machine cooperation research, which is concerned with strategic and tactical communication between humans and machines [87; 88; 86]. In this chapter, we will survey uses of human-AI shared control in six application domains and introduce the fundamental design considerations of human-machine cooperation.

## 4.1 Human-AI Shared Control

This chapter is concerned exclusively with *human-AI* shared control, in which a human user shares control with an AI agent. Human-AI shared control is not the only form of shared control. For example, human-human shared control has been explored as a means to improve the accessibility of sailing [9; 8; 15], skiing [7], and web browsing [229] to users with disabilities; as a way to mediate crowdsourced work [178; 176; 177]; and as a game mechanic [187; 109; 110]. However, due to our special interest in sharing control with AI, such as in partial automation, human-AI shared control will henceforth be called *shared control*.

In shared control, the human user controls a system to perform their tasks and an AI agent provides assistance. This interaction paradigm is used not only in gaming applications, such as partially automated games, but also several other application domains in which human users may have difficulty controlling the system. Since both control the system cooperatively, the AI serves to *extend* the human's

capabilities, *relieve* them of burdens, or *partition* and perform part of their functions [273; 137; 274]. For example, aim assist—a form of player balancing that helps players aim in first-person shooter games [306]—can extend players’ capabilities by helping them to hit targets that might otherwise be too small to aim at precisely. Both the human player and the AI copilot perform the task of aiming, which helps the player to aim more effectively. In contrast, Tesla’s *Navigate on Autopilot* self-driving feature fully automates highway driving, thereby relieving the driver of all operational tasks [294]. Drivers using this feature still need to monitor what the AI is doing and prepare to take control should the need arise, but they no longer need to use the accelerator, brakes, or steering wheel while it is engaged. As a final example, *Origin* is a handheld woodworking router that adjusts the position of its drill bit to help the user more precisely trace shapes [253]. This decomposes the user’s task into two complementary parts: the AI ensures that the shape is cut correctly and the human positions the router over the cutting area. In these ways, human-AI shared control can make users’ interactions more accessible, safe, precise, reliable, creative, and playful.

In shared control, two *actors*, one human and one AI, control a system through a *common interface* [72; 213; 214]. For example, power wheelchairs are often controlled by a joystick (Figure 3.3), which can be difficult to accurately manipulate by persons with fine-motor deficits [248]. In a *smart* power wheelchair, an AI agent provides its own joystick commands and works with the user of the wheelchair, adapting the user’s commands to avoid collisions and to better navigate to a desired destination [81]. Both actors control the system directly and in real time. The sequence of commands that each actor provides to the system is called a *control*

*signal* [213; 214] and both are used to construct a new *shared control signal*. This creates a tightly-coupled form of cooperation between human and AI that unifies their control of an interactive system. In contrast, Apple’s Maps application suggests destinations based on the contents of the user’s calendar. This example is *not* shared control, because it is not tightly-coupled, and because the AI agent does not share a control interface with the user; it makes suggestions, but does not actually control the car. In the next section, we describe how shared control can improve users’ interaction in six application domains.

#### 4.1.1 Shared Control Domains

There are many ways in which shared control can improve users’ interactions with computers. For example, as we saw in Chapter 3, digital games use shared control to improve their accessibility to players who have difficulty using controllers and making gameplay decisions. In this section, we will describe how shared control can also improve human work in five other application domains. Teleoperated robots and unmanned aerial vehicles (UAVs) share control to make complicated manipulation tasks easier [118; 92; 139; 148; 147; 71; 70; 72; 248; 76; 69]. Surgical robots filter out tremulous movements and guide the surgeon’s control of their implements [141; 117; 291; 152; 192; 239; 151; 322]. Creativity support systems make sketching and playing musical instruments easier using motors or electrical muscle stimulation (EMS) [327; 297; 159; 158; 290; 293; 271; 277; 191]. Semi-autonomous vehicles help drivers to stay in their lanes and prevent dangerous maneuvers [286; 282; 149; 310; 272; 174; 294; 93; 89]. Smart power wheelchairs and mobility assistance robots help users to navigate smoothly and

safely [314; 184; 67; 81; 80; 188; 189; 279; 99]. In this section, we survey uses of shared control in six domains: *digital games*, *telerobotics*, *surgery*, *creativity support*, *semi-autonomous driving*, and *mobility assistance*.

#### 4.1.1.1 Digital Games

As explained in Chapter 2, digital gaming is an enormously popular pastime that disadvantages or even excludes players due to differences in ability [329]. Shared control is used in some games to make them accessible and to level the playing field between persons with different abilities [52]. As explained in Chapter 3, shared control in games is called partial automation and it comes in two forms: input automation and one-switch automation.

Even among games that provide the same type of partial automation there is variability in the roles that the AI plays and the ways in which it assists players. For example, partial automation designed for player balancing helps weaker players compete with stronger players [17] and the copilot in *Gekku Aim* [134] aims directly at the closest opponent when the player shoots, stepping in at the last moment to override the player’s command. The player and the copilot both aim, so the copilot’s role is more supportive than complementary. In comparison, acceleration automation in *Mario Kart 8 Deluxe* [219] makes the player’s kart accelerate constantly, as though the player is holding down the “A” button. Since the copilot is given exclusive control of the kart’s acceleration, the player is relieved of this task entirely. Both of these games use input automation to improve their accessibility, but the reasons why and the ways in which the copilot assists players are different.

Many games using partial automation are classified as *one-switch games* and were developed to make existing games more accessible to players with motor disabilities. For example, *Alienated* [244] is a one-switch clone of the popular arcade game *Space Invaders*. The copilot automates control of shooting lasers at enemies and pressing the button makes the player’s avatar move right, if it was moving left, or left, if it was moving right. In this way, one-switch automation overcomes a pervasive problem encountered by players with motor disabilities, extending access to the fully-featured games enjoyed by their non-disabled peers [52]. Assisting human users who have difficulty controlling a system is common to all six of the domains described in this section, although automating control of inaccessible inputs is unique to games.

#### 4.1.1.2 Semi-Autonomous Driving

Fully autonomous vehicles are fast approaching, but in the meantime drivers have been invited to share control with driving automation systems. Here, both the human and AI control the car through its interface of accelerating, braking, and steering. These systems come in two types with similar acronyms: advanced driving assistance systems (ADAS) and automated driving systems (ADS) [228].

ADAS assist drivers in performing *primary* [243; 180] or *operational* [228; 204] driving tasks, such as accelerating, braking, and steering. For example, lane-keeping assistants (e.g., [286; 282; 149; 310; 272]) help drivers to stay in their lane by steering towards the lane’s center when the vehicle deviates. This leads to tightly-coupled interaction, where the human and AI agent can be literally moving the steering wheel at the same time, the human with their hands and the agent with a

motor. ADAS' assistance helps drivers to drive more safely when they are fatigued or distracted, and the AI's intervention can help them to recognize their mistakes, for example changing lanes without signalling [310].

In contrast, ADS relieve drivers of driving tasks entirely [174; 294]. For example, Tesla's Navigate on Autopilot fully automates highway driving from on-ramp to off-ramp [294]. The human is required to drive before the feature is initiated and after it is disengaged, as well as supervise the AI at all times so that they can take over in case of an emergency. ADS systems remove much of the tedium from driving, since they relieve drivers of their tasks by delegating them to the AI, and have the potential of increasing the safety and accessibility of driving [283; 27; 108; 107; 37; 131; 132; 38; 320; 36; 35; 34; 82; 33].

Much like systems from other domains, driver support systems work to prevent disaster when control is difficult, although some are able to support drivers even further. They may use *haptic shared control*, which provides the additional benefit of communicating the AI's intentions to the driver via force feedback [3]. For example, Volvo's Lane Keeping Aid rotates and vibrates the steering wheel when the vehicle unexpectedly crosses over lane markings [310], thereby informing the driver of their situation and amending their control. This same approach is used in surgical robots to improve surgeons' awareness of the tissue they are cutting [240]. Otherwise, driving automation systems relieve drivers of their tasks entirely. This use case is unique to semi-autonomous vehicles, since driving is the only application domain we surveyed in which it is preferable for the AI to do all the work. In contrast, input automation controls inaccessible inputs to make games more accessible, not to make play less tedious.

#### 4.1.1.3 Creativity Support

Shared control can also support humans in creative endeavors such as making sketches or sculptures [327; 297; 331; 158; 191; 159], musical performances [290; 293], and digital artifacts such as levels for a digital game [277] or labelled datasets [271].

For example, *dePEND* [327] is a sketching assistant that helps designers to draw shapes by guiding a pen across a sheet of paper using a magnet. By drawing two short lines at two points on the sheet, users can instruct *dePEND* to drag the pen from the second point to the first. The AI therefore supervises the human while they sketch; however, the user is free to wiggle the pen, to create wavy lines, or periodically lift the pen to create dotted lines. The user requests assistance by modifying the artifact and *dePEND* responds to these changes. Both human and AI control the pen at the same time in the real-time task of drawing. The collaboration is tightly-coupled, where movements of the pen immediately convey intent.

Systems can also assist users in real time performances, such as playing an instrument. For example, *PossessedHand* [290] uses EMS to share control of the user's body while playing a traditional Japanese instrument called the *koto*. Fourteen electrodes are placed on the user's forearm and EMS causes the user's hand to play along with a piece of sheet music. In this way, *PossessedHand* demonstrates to users how the *koto* is played and may facilitate learning to play without assistance. These systems were designed to overcome technical barriers experienced by creators, guiding their performance of difficult tasks, and are therefore especially useful to novices.

Shared control in creativity support is often explicitly didactic. *PossessedHand*, for example, was designed for users who had never played the *koto* before. In

contrast, surgical robots guide surgeons' control of their implements to supervise and support them, but not to teach them how to perform an operation. This makes creativity support unique among domains we surveyed in that, although it is not yet known whether using such a system can yield long term improvement in the performance of creative acts, creativity support AI can be designed to show novice users how a task should be done.

#### 4.1.1.4 Mobility Assistance

Many people have mobility disabilities, as did an estimated 9.6% of Canadians in 2017 [285], and may need the assistance of a device, such as a walker or wheelchair, to get around. Mobility assistance devices enable many users to navigate independently, but others may find using their wheelchair's joystick too difficult. These users may be able to use an alternative device, such as a sip-n-puff or head controller, but these devices may themselves be too difficult or tiring to use without assistance [188; 80]. Therefore, smart power wheelchairs share control with their users, helping them to avoid obstacles and navigate smoothly. For example, Soh & Demiris designed a smart power wheelchair [279] that learns and mimics the *hand-over-hand* assistance occupational therapists provide for novice power wheelchair users.

Mobility assistance devices use shared control to assist users who have difficulty navigating independently. Therefore, their designs embed the assumption that users need to be supervised to navigate safely. All of the devices we surveyed help users to avoid collisions [314; 184; 67; 81; 80; 188; 189; 279; 99] and many help with navigating smoothly [314; 184; 67; 81; 99]. For example, Ezeh et al. compared

two smart power wheelchairs [80], representing radically different forms of shared control. The first implements *linear blending* (i.e. *policy blending* [72] or *direct blending* [227]), a form of shared control that weights and sums the human user and AI agent's control signals. The path planner generates a control signal that smoothly steers the wheelchair away from obstacles and uses the average of the human and AI's control signals to control the wheelchair. In so doing, the linear blending wheelchair supervises the user and refines their control signal. In contrast, the second smart power wheelchair implements *probabilistic shared control* [80; 81], a form of shared control that learns from the human to infer their intentions [299]. The path planner uses its model of the human to select a command that it judges the human will find most agreeable and that best satisfies obstacle avoidance and smoothness constraints. The probabilistic approach used by this wheelchair illustrates another assumption embedded in the design of mobility assistance devices using shared control: since the human is expected to err, it is better to interpret their intentions than obey their commands. In this way, many systems use the AI to supervise and override the human's control [314; 184; 67; 81; 80; 188; 189], preventing disaster when the human is unable to navigate safely.

Much like ADAS in cars, shared control mobility assistance devices use the AI to supervise the human's actions and assist when their actions are dangerous. However, these systems typically interpret the human's commands in a way that is specific to their problem domain, and may be inappropriate in other situations. For example, the second smart power wheelchair described in the previous paragraph gives the AI more authority than the human user. The AI interprets the human's

intentions and navigates on their behalf. In contrast, surgical robots have been designed to filter surgeons' tremors, but it would be inappropriate for these robots to operate autonomously because the surgeon is considered the ultimate authority [240]. Differences in the design of these systems expose designers' assumptions about each actors' capabilities and how their interactions should be structured. Therefore, shared control of mobility assistance devices takes the form of a strict hierarchy, with the supervisory AI interpreting the human's commands and acting on their behalf from above.

#### 4.1.1.5 Telerobotics

The first shared control systems were designed to make dangerous and difficult work easier for humans [72]. In particular, teleoperated robots can perform manipulation tasks on the human's behalf [248; 71; 70; 72; 139] or explore environments that are unsafe for humans [76; 147; 148; 92; 69; 118]. For example, Rakhimkul et al. created a robot arm for users with motor disabilities that identifies objects in its environment and changes its pose to make it easier for the human to pick them up [248]. Therefore, telemanipulation systems use shared control to make human work and activities of daily living safer and easier. Otherwise, teleoperated robots can work in places that are inaccessible or unsafe for humans. For example, recreational drones—miniature unmanned aerial vehicles—such as DJI's Mavic Air 2 [69], or those created for research (e.g., [76; 147; 148; 92; 118]), could assist search and rescue services in locating persons missing in dangerous or difficult environments.

Teleoperated robots can automate tasks at a distance and in environments that would be unsafe for humans. But, as has been demonstrated time and time again, automation does not replace human work, it changes human work [14; 237]. And so, telerobotics comes with its own set of problems that shared control is used to overcome. For example, telemanipulation AI can share control of a robot arm that the user controls using a joystick [248] or skeleton tracking system [71; 70; 72]. Using these input devices may be tiring, imprecise, or difficult, so the AI assists to make the human's control less arduous. In some cases, working remotely also introduces issues related to signal interference and time delays that shared control can alleviate. For example, the Mavic Air 2 has a Failsafe Return to Home feature that causes it to fly back to a specified location when the human's control signal is lost for more than three seconds and Engel et al. created quadrocopter AI that overcomes unstable behaviour caused by short time delays [76]. Shared control likely cannot overcome problems caused by extreme time delays, such as those experienced by robots on Mars [300], but it can account for short lapses in real-time control. Therefore, shared control is used in telerobotics to overcome some issues inherent to controlling robots remotely, making them more reliable.

Shared control telerobots leverage AI to make the human's tasks easier and safer, but in ways unlike systems in other domains. For example, shared control telerobots expect interruptions in the human's control signal, but fewer than half of the systems we surveyed used these data to construct the AI's control signal (i.e., [76; 147; 148; 92; 69]), as is typically done by mobility assistance AI. Therefore, telerobots are designed to provide a form of shared control characterized by the human and AI's real-time and joint performance of their tasks. Rather than supervising

the human and trying to infer their intentions, telerobots share control by performing the same tasks as the human and providing continuous support, working in the background in case the human needs assistance.

#### 4.1.1.6 Surgery

Surgery is a risky and delicate task that sometimes requires more precision than human surgeons can provide. As explained by Jakopec et al., total knee replacement surgery leads to large misalignment of knee prostheses, which may necessitate revision surgery in over one third of cases [141]. It is surgeons' need for precision that drives the development of shared control surgical robots [240]. These are grounded robots (i.e., affixed to a static structure, such as an operating table) or ungrounded robots (i.e., mounted on the surgeon's body or tools) that share control of the surgeon's implements. For example, the grounded Acrobot guides the surgeon's control of an orthopaedic cutter system for milling patients' bones during total knee replacement surgery, eliminating unwanted deviations in prosthesis alignment [141]. In this way, shared control surgical robots overcome imprecision in surgeons' control of their implements, making their work safer and more effective.

The imprecision that surgical robots are typically designed to overcome is called *tremulous motion*. These are involuntary and high-frequency hand movements that may be unnoticeable in non-surgical contexts but disabling for surgeons, as they can be orders of magnitude larger than some of the smaller bodily structures surgeons routinely manipulate [192]. Due to their distinctively high frequencies,

surgical robot designers rely on a signal processing analogy in which shared control is used to “*filter*” [192; 240; 291; 152] tremors. The ungrounded Micron system [192], for example, implements a low-pass filter that removes high-frequency components from the surgeon’s control signal, using only the low-frequency components to drive a piezoelectric manipulator that moves the tool’s tip. This can give the surgeon the sensation of being in complete control while the assistant removes tremulous motion that the surgeon may not even be aware of.

These systems enable surgeons to operate with superhuman precision and their motors can confer superhuman awareness as well. For example, instead of reducing noise, the force amplifying device developed by Payne et al. amplifies forces at the tip of the surgeon’s implement [239]. This enables the surgeon to determine an appropriate amount of pressure to apply, overcoming the reduced kinaesthetic feedback surgeons experience during minimally invasive surgery [240]. A similar approach was used in Acrobot, which uses force feedback to prevent surgeons from cutting beyond predefined regions. In this way, surgical robots can share control of the surgeon’s implements to guide their control, informing them of their mistakes and preventing them from making them.

Designers’ expectations of surgeons’ capabilities make surgery unique among domains we surveyed. The human user in these systems is a surgeon who is expected to be less precise than the AI, but far more knowledgeable and prudent. Games, mobility assistance devices, and telerobots expect that human users may be unable to provide their intended command, so they step in to control the system on the user’s behalf. Some creativity support systems guide users’ performance and even go so far as to force the user to take action when they may not have intended

to. In contrast, surgical robots expect their users to err, but never act on the human's behalf. They have been designed to filter out a specific form of noise in the human's control signal, leveraging shared control to discover the *true signal* that it suggests. Force feedback can inform surgeons about the tissue they are cutting and help them to recognize when they have made a mistake, but it is never used to force the surgeon's hand. Therefore, shared control is used by surgical robots to support surgeons' control of their implements as a subordinate, but not as an equal or superior.

We see from this presentation the great variety of cooperation styles shared control can afford. When control is shared, AI can assist humans with many different types of tasks and in ways specific to each task's demands and each actor's capabilities. We saw how smart power wheelchairs and surgical robots both work to prevent users from making errors, but also how these systems' designs embed domain-specific assumptions about which actor needs to be supervised and who has the authority to act unilaterally. Telerobots often perform the same tasks in concert with users, while semi-autonomous vehicles might allow users to delegate their tasks to the AI. Some creativity supports guide users' control using motors or EMS while partially automated games more often interpret players' commands to make controlling the game easier. Designers of partial automation can take inspiration from uses of shared control in these other domains when considering what roles the copilot should perform and how it should cooperate with the player. We will revisit these considerations in Chapter 5 when we present our dimension space intended to facilitate the design of shared control systems. Now, we turn our focus

towards the broader topic area of human-machine cooperation and discuss more general considerations relevant to the design of shared control systems, such as partial automation.

## 4.2 Human-Machine Cooperation

While shared control is concerned primarily with cooperation at the operational level (i.e., control), *human-machine cooperation* is concerned with cooperation at the strategic and tactical levels; that is, deciding what to do and how to do it. For humans to cooperate effectively with machines, they need to understand how to use them and be able to predict what they will do. But that is not enough. Machines also need to, in some sense, understand users' goals and both actors need to communicate to coordinate their actions. When designers broaden their considerations beyond who does what, they enter a design space where users' perceptions of AI [236] and the social context in which they cooperate [10] are more salient. Machines can perform many types of human tasks at various levels of authority commensurate in their capabilities, including: the acquisition and analysis of information, the selection of decisions, and the implementation of actions [237]. But one of the most fundamental [14] and persistent [284] ironies of automation is that automating the simple and routine parts of human work can make the remaining work more complicated and variable. In this section, we discuss the human factors of human-machine cooperation relevant to the design of partial automation.

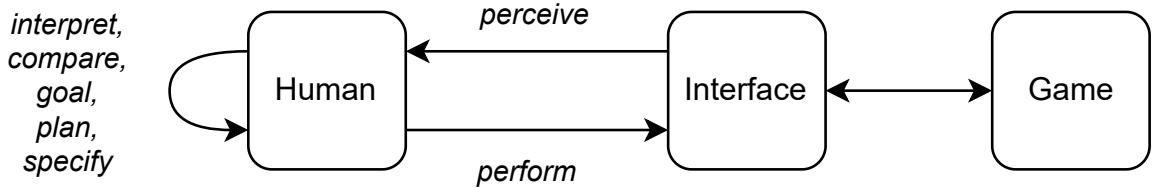


Figure 4.1: Norman’s seven stages of action grounded in the game interaction model introduced in Chapter 2. After forming a *goal*, users *plan* how they will achieve it and *specify* the sequence of actions needed to enact their plans. Users then *perform* the action sequence and *perceive* how their actions affect the state of the world. Finally, users *interpret* their perceptions and *compare* their interpretations to their goals to determine whether they were achieved. The *perceive* stage corresponds to *receive stimuli*. The *interpret, compare, goal, plan, and specify* stages correspond to *determine responses*. The *perform* stage corresponds to *provide input*.

#### 4.2.1 Knowledge & Understanding

When interacting with a system, users form in their minds models of how the system operates, called mental models [220; 221; 307; 160; 241]. Users’ interactions follow seven stages of action (Figure 4.1), from specifying a goal to determining whether it was achieved, and each step along the way depends on the quality of the user’s mental model. Good mental models enable users to recognize and understand what the system is doing, crossing the *gulf of evaluation* (i.e., *perceive, interpret, & compare*), and also to predict how their use of the system might achieve their goals, crossing the *gulf of execution* (i.e., *plan, specify, & execute*). Bad mental models make users confused about what the system is doing and how it might respond to their actions. Prior investigations into the human factors of highly automated flight decks found that pilots’ most commonly asked questions about the automation were: “*What is it doing?*”, “*Why is it doing that?*”, and “*What will it do next?*” [264; 319] These sorts of questions are answered by users’ mental models [220; 221; 270; 150].

Human-machine cooperation decomposes users' knowledge of a joint task into their *know-how* and their *know-how-to-cooperate* [232; 207; 87]. Know-how comprises users' knowledge of how to operate a system, which necessitates good mental models. Know-how-to-cooperate comprises users' knowledge of their collaborators, including their capabilities, intentions, and how to coordinate with them.

Rather than providing unified and complete explanations of all aspects of a system's operations, the mental models users form through use are fragmentary and incomplete [270]. They are collections of theories analogous to specific parts of the system and they can leverage metaphors to explain how and why the system does what it does [58]. *Functional* mental models tell users how to use the system, while *structural* mental models tell users how the system works [172]. While investigating users' mental models of calculators, Norman found that users would press the 'CLEAR' button excessively due to erroneous beliefs about how the calculator stored values in memory [221]. These erroneous beliefs led them to develop harmless "*superstitions*" about the effects of their 'CLEAR' presses.

#### 4.2.2 Superstition & Surprise

Although automation can improve users' interactions, it can also make users' tasks more difficult and complicated [14; 284; 319; 222]. The automation may perform tasks that the human does not know how to do using knowledge that the human has not learned. Therefore, users may be unable to understand the automation's actions and have difficulty cooperating with it. The mode of an automated system determines how it responds to users' input [143] and *mode confusion* occurs when users provide inappropriate input because they are unaware of the automation's

current mode [266; 264; 263; 265]. Users encounter *automation surprises* when the automation does something unexpected, which can lead them to develop *curious rituals* (e.g., tilting controllers, hitting hardware, etc.) that only appear to cause the desired behaviour [223]. Throughout the rest of this thesis, we refer to seemingly confused understandings such as these as “superstitions”, which we define as illusions of causation. Humans have a tendency to accept evidence that confirms their beliefs [164] (i.e., confirmation bias) and also to infer causal relations from co-occurrences [325; 21], so automation that is outside the user’s control may promote illusions of control [175]. To err is human; when automation is designed without consideration for human fallibility, users may come to believe in things that they don’t understand and suffer the consequences.

For example, drivers were expected to change their braking behaviour with the introduction of anti-lock braking systems (ABS), which pumps the brakes for the driver to prevent wheel lockup. However, survey data from 1998 indicates that 18% of ABS users thought that they had to pump the brakes to activate ABS [42]. In 2017, the Boeing 737 Max entered service, equipped with a new Maneuvering Characteristics Augmentation System (MCAS) [235]. Pilots were not trained on how to use, or even told about, the MCAS, which would force the plane’s nose down when a single sensor detected that the plane’s pitch was too high [235; 280]. When faulty sensors caused the MCAS to activate erroneously during Lion Air flight 610, the pilots were surprised and unable to disengage the automation [280]. Automation accidents such as this (e.g., Three Mile Island [301; 242], Therac-25 [182; 266], Sudden Unintended Acceleration [151; 168]) demonstrate how automation can cause users to hurt themselves and others in their confusion.

### 4.2.3 Trust & Awareness

Users employ different combinations of analytic, analogical, and affective reasoning to determine when to trust automated systems, which may lead users to rely on them inappropriately [179]. Users may misuse the automation, applying it to tasks that it cannot do, or disuse the automation, not applying it to tasks that it can do [236]. A mismatch between the user's understanding of the automation and its actual operations means that when things go wrong, users may be surprised and not know how to respond [266; 264; 263; 60; 59]. To help users make sense of the automation, special interfaces have been developed.

Automated systems need to support users' awareness of automation to help them develop sound mental models. Insufficient awareness of automation can diminish users' task performance, as users may need to think harder to understand what the automation is doing [75]. Wintersberger et al. designed augmented reality aids, displayed on the windows of an autonomous vehicle, indicating the presence and proximity of other vehicles [321]. Other interfaces framed the automation as a collaborator that would inform users of its actions. Koo et al. created voice alerts to inform autonomous vehicle passengers that the “*Car is braking*” [167] and Häuslschmid et al. designed a chauffeur avatar that appeared to react to objects and drive the vehicle [135]. In doing more complex tasks, such as air traffic control, users' awareness of automation can be supported using a *common work space* that displays the user's situation along with the automation's current actions and intentions [207; 232].

Designers of automated systems can also learn from human-to-human collaboration through groupware. In groupware, information about collaborators' activities

is communicated via *awareness cues* [121]. As with sharing control with automation, cooperative work in groupware can be tightly coupled, meaning that users need to interact frequently to achieve their goals [261]. Awareness cues support tightly coupled interactions by conveying useful information about collaborators' interactions in a shared workspace, called the *elements of workspace awareness* [120]. For example, *Action* cues indicate what action a user is doing while *Authorship* cues indicate who is doing an action. The feedback that awareness cues provide may help users to attribute changes in the workspace to their collaborators and make sense of what they are doing.

In multiplayer games, awareness cues are used to help players coordinate with other players. Bortolaso et al. used awareness cues to facilitate communication between tablet players and VR players in an asymmetric home decoration game [28]. Stach et al. found that information-rich avatar embodiments helped players to more effectively strategize in an arcade-style space shooter [281]. Toups Dugas et al. proposed a framework of cooperative communication mechanics that facilitate non-verbal communication between team members [298] and Wuertz et al. proposed a framework for the design of awareness cues in games [326]. Awareness conferring game mechanics have been shown to be useful for enabling human-to-human cooperation in digital games, but it is unknown whether they can facilitate cooperation between a human player and automation.

## 4.3 Summary

Partial automation is a form of human-AI shared control and in this it is not alone. Shared control is used in at least five other application domains to overcome domain-specific problems. Designers of partial automation can take inspiration from these disparate uses and should give consideration to the human factors of automation discussed in this chapter. Although we have described many ways in which sharing control with AI can assist users, novel automation may sometimes hurt users more than it helps them. Sharing control can simplify users' tasks at the cost of introducing new tasks related to monitoring the automation, interpreting its behaviour, and predicting its future behaviour. For users who may never have performed the tasks that the AI automates, making sense of its behaviour may be intractable. In the second part of this thesis, we will present evidence that partial automation can improve the accessibility of digital games to non-gamers and players with motor impairments, but we will also discuss how sharing control with partial automation can make players confused.

This chapter concludes part 1. In this part, we have identified the reasons why non-gamers and players with motor impairments choose to play games and the barriers to access that they may encounter, described the approaches designers can take to improve a game's accessibility to these players, introduced the idea of partial automation, and surveyed the uses and human factors of human-AI shared control. In Chapter 5, we propose a dimension space that is informed by the systems surveyed in this chapter and intended to facilitate the design of human-AI shared control systems. In Chapter 6, we will propose a novel and personalizable form of partial automation and show how combining it with the universal design and interface

adaptation approaches described in Chapter 3 can make exergames accessible to players with a range of motor impairments. In Chapter 7, we will revisit the human factors issues described in this chapter, show how partial automation might confuse non-gamers, and propose a grounded theory intended to help designers identify and address players' confusion. Finally, in Chapter 8, we will propose a reference architecture and associated development toolkit to facilitate the design, and inevitable redesign, of partial automation.

## **Part II**

# **Research in Partial Automation**

# CHAPTER 5

---

## Shared Control Design Space Analysis

---

In Part I of this thesis, we identified two groups of players, *non-gamers* and *players with motor impairments*, who encounter significant barriers to access in digital games and we claimed that a novel accessibility feature, partial automation, may make games more accessible to these players. Non-gamers may want to play games with their friends and family, but most games expect players to have more gaming expertise than novices possess. Non-gamers may become frustrated when playing fast-paced games that require them to quickly make sense of what is happening onscreen and require them to rapidly manipulate a complex game controller. Players with motor impairments may want to play games for their own enjoyment, as a way to connect with others, or while performing routine exercises as part of their rehabilitation regimen. However, games place physical interaction

demands on players—most games require players to precisely tilt up to two analog sticks and rapidly press buttons on both sides of the controller simultaneously—and players with motor impairments may not be able to do so. If there is even one part of the game that a player cannot control, then they may be unable to play at all.

In Chapter 3, we explained that there are several approaches designers can take to improve the accessibility of their games to these players. Games can be made accessible to players with specific abilities from the outset (i.e., universal design), games can support multiple types of customizable controllers (i.e., interface adaptation), and games can assist less proficient players by reducing the difficulty of challenges presented to them (i.e., player balancing). However, we also claimed that the efficacy of these approaches have limits that partial automation can extend. In Chapter 4, we explained that partial automation is a form of human-AI shared control, similar to other shared control systems used for diverse purposes such as mobility assistance and computer-assisted surgery. We claimed that design insights from these disparate problem domains could inform the design of partial automation and we suggested, in reference to known issues of human-machine cooperation, that partial automation may also bring about new issues. Partial automation abstracts away a game’s complexities related to deciding what to do and controlling the game to make that happen. Players who have never played without partial automation might have difficulty understanding what is happening in the game and recognizing which in-game actions were caused by them and which were caused by the copilot.

This part of the thesis provides evidence to support these claims. It demonstrates partial automation’s capacity to improve the accessibility of digital games to

non-gamers and players with motor impairments, but also its potential to confuse players about how games are controlled. Furthermore, it proposes methods that designers can employ to identify and address players' confusion. Ultimately, we will argue that the design of partial automation can be facilitated by a reference software architecture and an associated development toolkit. In this chapter, we will address our claim that the design of shared control systems further afield can inform the design of partial automation. We will precisely characterize the forms of interaction afforded by the shared control systems described in Chapter 4 and identify common design patterns that can be used to analyze design choices. The primary research question addressed through this analysis is **RQ1**: “*What are the dimensions for design and design patterns supporting human-AI shared control?*”

## 5.1 Shared Control Design

We see from our presentation of the six application domains in Chapter 4 that human-AI shared control has arisen in numerous contexts, often independently and without knowledge of its use elsewhere. Each system is characterized by the human and the agent sharing the same interface—sometimes through software, and sometimes through physical embodiment of the agent through magnets, motors, or even electrical stimulation of the user's muscles. This usage of the system is tightly-coupled, often involving simultaneous control of the system by both the human and an AI agent. In partial automation, we call this agent the copilot and design it to cooperate in ways that players like. Our review in Chapter 4 has revealed some of the difficult decisions that are faced by designers of shared control systems. For example, should the human or the agent have primary control? What

forms of supervision should be present between human and agent? How and when should control transfer between the two actors? These are questions that the dimension space—a design tool to structure a design space—proposed in this chapter is intended to address.

Even though many technologies use shared control, the styles of interaction that they afford can differ vastly across domains. Designers working in one domain may be unaware of how shared control is used by others and may be unaware of solutions that have proven useful for solving similar problems. The terminology frequently differs, making it difficult for design insights discovered in one domain to be transferred to others. In order to understand how commonly used design patterns overcome a problem and to make comparisons between designs, we need a common language for describing the design space of shared control systems. We need tools that help designers to better communicate their ideas about how interactions between human users and AI agents should be structured in shared control, to identify gaps in the approaches used in one domain, and take inspiration from approaches used in others.

Design space analysis is “*a perspective on design which emphasizes the role and representation of design rationale.*” [193] It recognizes that a design process produces not only a reified artifact but also an abstract design space of possible options. This notion of design space analysis is closely related to Alexander’s notion of *pattern languages* for describing *design patterns* [6; 5]. The connection is most obvious in Alexander’s own words: “*The real work of any process of design lies in this task of making up the language, from which you can later generate the one particular design.*” [5] A design pattern is “*a rule which describes what you have to do to*

*generate the entity which it defines*" [5] and they can be composed to express novel designs using pattern languages.

There are at least two pattern languages that can be applied to shared control, although neither is specific to it. Baltzer et al. proposed an interaction pattern language for human-machine cooperation defined in terms of the problem a pattern addresses, the solution that overcomes the problem, the consequences of using the solution, and example systems that implement it [16]. This tool can express how and why a particular design overcomes a problem, but as van Diggelen & Johnson have previously argued [303], pattern languages reliant on natural language descriptions lack uniformity and therefore make comparing designs difficult. Since this pattern language provides no guidance regarding the design space of shared control systems, designers using it could easily get lost. van Diggelen & Johnson have proposed their own pattern language for human-agent team design patterns, defined in discrete terms [303]. They considered the types of work actors perform, whether the work is physical or cognitive, actors' spatial distribution, and how actors communicate. Using these elements of team work, the authors have demonstrated how higher-level design patterns, such as human supervisory control—an interaction paradigm in which human users' interactions are mediated by automation that they supervise [273], could be expressed. This pattern language enables designers to describe existing human-agent team patterns, and envision new ones, but it cannot describe how teams share control. It indicates that understanding how actors supervise each other is necessary for understanding how they cooperate, but more specificity is needed to understand how design choices affect users' experiences.

Many recent works [87; 237; 181; 206; 90; 232; 207; 234] have iteratively constructed a human-machine interaction model composed of layers of shared and cooperative control, assistance, and automation [233]. As explained in Chapter 4, human-machine cooperation involves a human and AI agent communicating with each other and controlling a system, via a Common Work Space [232; 207]. This framework can express what types of tasks each agent performs [237], on what levels they communicate [232], and whether they control the system directly, but it cannot express how agents' control signals are composed to create a shared control signal. In a similar vein, Abbink et al. have proposed a design framework for shared control systems [2]. It can express how human and AI perform a hierarchy of tasks and communicate using signals, signs, and symbols [249] at each level, but says little about how a particular design choice might influence users' experiences. These design frameworks provide an especially technical account of shared control, and thereby provide little intuition for how designers' choices affect users' experiences of sharing control.

In this chapter, we attempt to elucidate a design space of shared control systems and propose a design space analysis tool through which it can be understood. The tool we constructed is called a *dimension space*; these are used to structure a design space, classifying and comparing systems along different *dimensions*. For example, one could construct a dimension space to understand the properties of interactive systems in a physical environment according to the system's role and the physicality of its manifestation [115], or to classify and compare musical devices according to their required expertise and number of degrees of freedom [26]. Dimension spaces enable designers to describe a design space, explore the design choices available to

them, and communicate their design rationale to others. When they describe meaningful differences between designs, they can help designers to realize possibilities that they may not have conceived of otherwise. To the best of our knowledge, no such tool exists for reasoning about the design of human-AI shared control systems. This is the gap that our dimensions space fills. For the first time, designers are able to express how design choices determine how human users interact with their AI partners. We provide a structure to the design space of shared control systems and describe how different design choices afford different interactions and experiences to users. In the next section, we describe our surveying method and how the dimension space was constructed.

## 5.2 Dimension Space

In order to better understand how human-AI interactions are structured in shared control, we set out to create a dimension space, which is shown in Figure 5.1. We compared descriptions, diagrams, use cases, and user evaluations of shared control systems to identify dimensions of variability that have meaningful consequences for the interactions they afford. We surveyed the literature using an approach similar to a scoping review [11]. Our goal was to discover the boundaries of shared control and identify problem domains in which it is used, so that designers working in one domain could understand design insights from other domains. We followed an iterative process in which we (1) searched online databases using keywords found in relevant papers, (2) identified papers that described human-AI shared control systems that were somehow novel, and (3) mined these papers for new keywords and references to potentially relevant systems (Table 5.1) We grouped

Table 5.1: The systems examined, topics encountered, and search terms used in each phase of our survey.

Phases	Keywords	Systems
<b>Phase 1</b> (Google Scholar, ACM Digital Library)	player balancing, input automation, crowd control semi-autonomous vehicles, haptic shared control, drones smart power wheelchairs, mobility assistance robots, human-supervisory control, aviation automation, smart homes, teleoperated robots, human-adaptive mechatronics, surgical training	(58 found, 45 excluded) <b>G1, G3</b> <b>D2, D3</b> <b>M2, M4-7, M9</b> R2, R6-8
<b>Phase 2</b> (IEEE Xplore, ScienceDirect)	human-machine cooperation, cooperative control, adaptive automation, adaptive cruise control, lane keeping assistants, air traffic control, quadrocopters, quadrotors, comanipulation robots	(41 found, 29 excluded) <b>D1, D4, D5, D7, D9</b> <b>M1, M3, M8</b> R1, R3, R4 <b>S8</b>
<b>Phase 3</b> (Google Scholar, ACM Digital Library)	interactive machine learning, gaze control, mixed-initiative, interactive fabrication, sketching assistants, electrical muscle stimulation, human-robot teams, human-AI teams,	(57 found, 40 excluded) <b>G2, G4-8</b> <b>D6, D8</b> <b>C1-4, C6-10</b> R5, R9
<b>Phase 1 Bis</b> (Google Scholar ACM Digital Library)	surgical robots, <i>all of the above</i>	(8 found, 1 excluded) <b>S1-7</b>

systems according to the problems that they were designed to overcome and called these groups *domains*. We then compiled a list of the systems we encountered in these domains and called this our *corpus* (Table 5.2). In the rest of this chapter, these systems will be referenced by their ID in Table 5.2.

As we iteratively built our corpus, we described the systems we encountered and recorded how these systems' designs differed. By considering what made systems similar or different informally and subjectively, we constructed dimensions of variability that enabled us to classify systems more formally and objectively. Periodically we chose the dimensions that we believed were most important for explaining how human-AI interactions in shared control are designed, and classified all of the systems in our corpus using these dimensions. When our dimensions classified systems as inappropriately similar or needlessly different, we split or combined axes to form new concepts. This subjective process of expanding our dimension space to capture notions inspired by the literature, and contracting our dimension space by

Table 5.2: Our corpus of shared control systems according to their classifications in our dimension space. More intense colors indicate greater extension along an axis. Italicized names are those designated by the system's designers.

Domain	ID	Name	AI Role	Supervision	Influence	Mediation
Games	G1	Zac - O Esquilo [203]	Complementary	Unsupervised	Independent	Combined
	G2	<i>Alienated</i> [244]	Complementary	Unsupervised	Independent	Combined
	G3	<i>Personalizable Partial Automation</i> [52]	Complementary	Unsupervised	Independent	Combined
	G4	<i>Imaginary Pong for One</i> [190]	Complementary	Unsupervised	Guided	Combined
	G5	<i>Mario Kart 8 Deluxe</i> [219]	Delegated	By AI	Independent	Combined
	G6	<i>Gekku Aim</i> [134]	Supportive	By AI	Interpreted	Selected
	G7	Racing game [46]	Supportive	By AI	Interpreted	Combined
	G8	Missile Command clone [63; 65; 64]	Cooptable	By Human	Independent	Selected
Semi-Autonomous Driving	D1	Lane-keeping assistant [286]	Supportive	Unsupervised	Guided	Combined
	D2	Lane-keeping assistant [282]	Supportive	By Human	Guided	Combined
	D3	Lane-keeping assistant [149]	Supportive	Mutual	Codependent	Combined
	D4	<i>Lane Keeping Aid</i> [310]	Supportive	Mutual	Codependent	Combined
	D5	Lane-keeping assistant [272]	Supportive	By AI	Interpreted	Selected
	D6	Automated driving system [174]	Delegated	By Human	Independent	Selected
	D7	<i>Navigate on Autopilot</i> [294]	Delegated	By Human	Independent	Selected
	D8	<i>Hotzenplotz Interface</i> [93]	Cooptable	Unsupervised	Independent	Selected
	D9	H-Mode car [89]	Cooptable	By Human	Codependent	Combined
Creativity Support	C1	<i>PossessedHand</i> [290]	Supportive	Unsupervised	Guided	Combined
	C2	<i>EMS Air Guitar</i> [293]	Supportive	Unsupervised	Guided	Combined
	C3	<i>dePEND</i> [327]	Supportive	By AI	Guided	Combined
	C4	<i>Haptic Intelligentia</i> [297]	Supportive	By AI	Guided	Combined
	C5	<i>Freed</i> [331]	Supportive	By AI	Guided	Combined
	C6	Sketching assistant [158]	Supportive	Mutual	Guided	Combined
	C7	Image classifier [271]	Cooptable	By Human	Guided	Selected
	C8	<i>Tanagra</i> [277]	Delegated	By Human	Independent	Selected
	C9	<i>Muscle-Plotter</i> [191]	Delegated	Unsupervised	Independent	Combined
	C10	<i>TransPen &amp; MimeoPad</i> [159]	Complementary	Unsupervised	Independent	Combined
	C11	<i>Origin</i> [253]	Complementary	Unsupervised	Independent	Combined
	C12	Digital Airbrush [275]	Complementary	By AI	Codependent	Combined
Mobility Assistance	M1	Mobility robot [314]	Supportive	By AI	Interpreted	Combined
	M2	Power wheelchair [184]	Supportive	By AI	Interpreted	Combined
	M3	Power wheelchair [67]	Supportive	By AI	Interpreted	Combined
	M4	Power wheelchair [81]	Supportive	By AI	Interpreted	Combined
	M5	Power wheelchair [188]	Supportive	By AI	Interpreted	Selected
	M6	Power wheelchair [189]	Supportive	By AI	Interpreted	Selected
	M7	Power wheelchair [279]	Supportive	By AI	Independent	Selected
	M8	Power wheelchair [80]	Supportive	By AI	Independent	Combined
	M9	Mobility robot [99]	Complementary	By AI	Independent	Combined
Telerobotics	R1	Telemانipulation robot [248]	Supportive	By AI	Independent	Combined
	R2	Telemانipulation robot [71; 70; 72]	Supportive	Mutual	Independent	Combined
	R3	Unmanned aerial vehicle [76]	Supportive	Unsupervised	Interpreted	Combined
	R4	Unmanned aerial vehicle [147]	Supportive	By AI	Interpreted	Selected
	R5	Rescue robot [148]	Supportive	By AI	Interpreted	Selected
	R6	Unmanned aerial vehicle [92]	Supportive	By Human	Independent	Combined
	R7	Telemانipulation robot [139]	Supportive	Mutual	Codependent	Combined
	R8	<i>DJI Mavic Air 2</i> [69]	Reciprocal	Mutual	Interpreted	Combined
	R9	Unmanned aerial vehicle [118]	Complementary	Unsupervised	Independent	Combined
Surgery	S1	<i>Steady-Hand</i> [291]	Supportive	Unsupervised	Codependent	Combined
	S2	<i>Micron</i> [192]	Supportive	Unsupervised	Codependent	Combined
	S3	<i>Craniostar</i> [151]	Supportive	By AI	Codependent	Combined
	S4	<i>Supervisory Steady-Hand</i> [152]	Delegated	By Human	Codependent	Combined
	S5	Comanipulation robot [322]	Complementary	Unsupervised	Codependent	Combined
	S6	Force amplifier [239]	Supportive	Unsupervised	Guided	Combined
	S7	<i>Acrobot</i> [141]	Supportive	By AI	Guided	Combined
	S8	Comanipulation robot [117]	Supportive	By AI	Guided	Combined

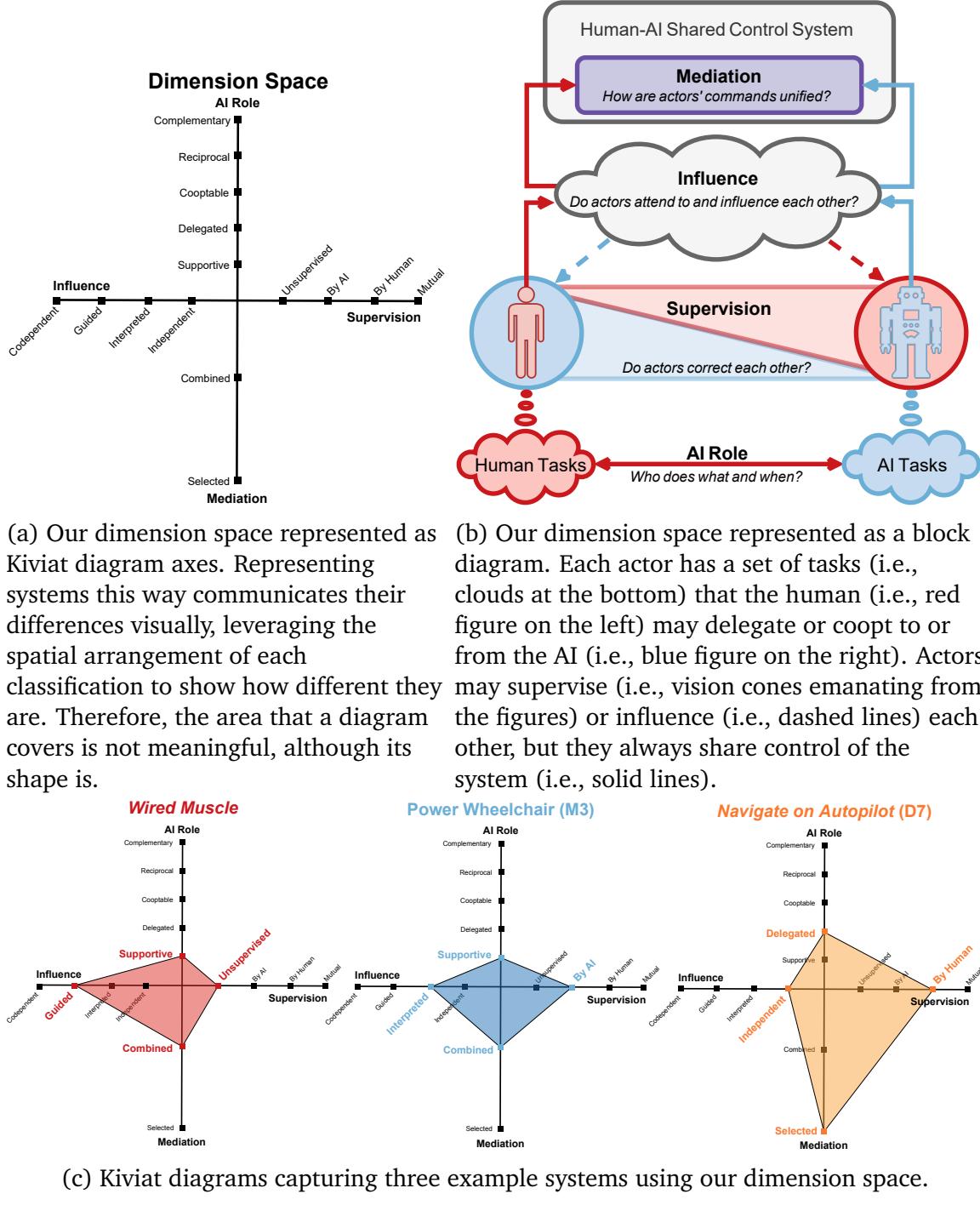


Figure 5.1: Visual representations of our dimension space.

removing and combining axes, was repeated until we converged on a set of dimensions describing what we believed to be most important similarities and differences for explaining human users' interactions with the systems we surveyed.

### 5.2.1 Dimensions

The dimension space has four axes: *AI Role*, *Supervision*, *Influence*, and *Mediation*. Each axis contains multiple discrete forms of human-AI shared control that differ along that axis. They are depicted as Kiviat diagrams in Figure 5.1, as is conventional for dimension spaces (e.g., [26; 115; 124]). *AI Role* describes the different ways tasks might be shared between human and the AI agent. *Supervision* specifies how actors monitor and correct each other. *Influence* captures how an actor chooses its own actions in response to those of the other actor. *Mediation* describes how actors' commands are unified through combination or selection. As suggested by Figure 5.1b, these dimensions correspond to different aspects of actors' interactions in shared control. *AI Role* is about actors' tasks, while *Mediation* is about their commands. *Influence* is about being aware of and responding to the other's actions, while *Supervision* is about taking action to correct the other. These dimensions have been constructed so that they can describe all human-AI shared control systems of which we are aware, and so that any position in the dimension space describes a non-empty set of plausible systems.

We designed this dimension space to provide the properties of *totality*, *orthogonality*, and *mutual exclusion*, which we define here. Each dimension was designed to be *total*, meaning that systems belong to at least one of the categories along each axis. This property holds of every system in our corpus. The dimensions are

designed to be *orthogonal*, meaning that a system’s classification in any dimension is independent of its classification along the others. Not all dimension spaces are orthogonal (e.g., [26]), but this property is desirable for our purpose because every position in the dimension space describes a valid system that designers could then implement. Figure 5.1 shows how each dimension corresponds to a different aspect of the human and AI agent’s interactions in shared control. There is no reason, for example, why a specific role would constrain choices around supervision, influence, or mediation. Finally, we designed the dimensions to be *mutually exclusive*, meaning that a system occupies at most one point in the dimension space. This property holds of all systems in our corpus. By aspiring to these properties, the dimension space can be used to classify systems in which one human shares control with one AI agent, and can be used to identify common design patterns between sets of systems. In this section, we describe the four dimensions of the dimension space: *AI Role*, *Supervision*, *Influence*, and *Mediation*.

### 5.2.1.1 AI Role—Who Does What and When?

The AI Role dimension answers questions of the form “*who does what and when?*”, a phrase borrowed from Inagaki [137]. Users interact with a system for some purpose; they have some set of tasks that they are performing [44]. When control is shared, the AI can support the human by cooperatively performing these tasks at the same time, the AI can take over control of the human’s tasks to perform them at a different time, or it can perform its own tasks that are separate from the human’s. This dimension captures how tasks are shared among actors and how the human assigns tasks. It tells designers how the human’s interactions with the AI are related

to the tasks they perform and how these tasks are shared between them. In all of the systems we surveyed, it is the human who allocates tasks and never the AI. We therefore define five types of AI role: *Supportive*, *Delegated*, *Cooptable*, *Reciprocal*, and *Complementary*, which we define in this section.

### **Supportive: *The AI assists the human with a subset of the human's tasks***

The human is primarily in control when the AI performs a *Supportive* role. The human is engaged in all parts of their overall task while the AI performs some tasks at the same time to help the user if they run into trouble. For example, Gurnel et al. created a comanipulation robot (S8) that helps surgeons to more precisely insert needles into tissue samples using virtual fixtures. While the surgeon moves the needle, five haptic guides exert forces on the needle to guide the surgeon towards the desired needle position and orientation. Systems designed such that the AI performs a *Supportive* role assist humans with their tasks while the human retains primary control.

### **Delegated: *Supportive & The human can hand over a subset of their tasks to the AI***

AI agents that perform a *Delegated* role relieve humans of specific tasks and perform them on their behalf. For example, human users may command Muscle-Plotter (C9) to do physical simulations using car designs that they have sketched and then graph the results using EMS. Digital game level designers can instruct Tanagra (C8) to fill in missing level geometry after they have made a change to the level. *Delegated* AI performs tasks that the human could do but prefers not to, or takes over control of tasks that the AI can perform more effectively.

**Cooptable: Supportive & The human can take over a subset of the AI's tasks**

In contrast to the *Delegated* role, which enables humans to delegate their tasks to AI, the *Cooptable* role enables humans to take over control of the AI's tasks. For example, Flemisch et al. designed a semi-autonomous vehicle (D9) according to the H-Metaphor [288; 91], which suggests that drivers' interactions with autonomous vehicles should resemble riders' interactions with horses. In horseback riding, riders can *loosen* their grip on the reins, thereby shifting control authority to the horse, or *tighten* their grip, to seize authority. In this way, the vehicle autonomously drives around a racetrack, albeit less skillfully than a human might, while the driver subtly takes control, tightening the rein, to assist the AI. In sum, AI agents performing a *Cooptable* role are primarily in control of their tasks, while the human is able to take over the AI's tasks when they see fit.

**Reciprocal: Delegated & Cooptable**

When the AI performs a *Reciprocal* role, the human can delegate tasks to the AI and take over tasks that the AI is controlling. For example, the Mavic Air 2 (R8) is a recreational drone with an array of operational modes. *Active Track* enables pilots to instruct their drone to follow them and record video, delegating this task to the AI. The pilot can then take back control by turning the feature off, but only so long as the drone receives their control signal. Should the Mavic Air 2 lose the human's control signal, due to interference from the environment, the *Failsafe Return to Home* feature causes the drone to return to a predetermined location. Once their connection is reestablished, the pilot is free to disable this feature and coopt the AI's control of the drone's

navigation. Thus *Reciprocal* shared control AI confers the benefits of both *Delegated* and *Cooptable*. The AI relieves users of their tasks and enables them to amend the AI's control of its own tasks.

#### **Complementary: Supportive & The AI has its own tasks that the human never performs**

A *Complementary* design partitions the human's task and allocates some parts to the AI in their entirety. This role can reduce the human's control burden when controlling the system is complicated, and can make systems more accessible when the human is unable to perform some tasks at all. For example, Shaper's Origin (C11) is a woodworking router that helps users to cut along a reference path more precisely. When using a typical router, moving the device's frame also moves the cutting tool, causing it to cut. In contrast, Origin separates control of the frame, which is moved by the human, from the tool, which is moved within the frame by the AI. In this way, the human is tasked with positioning the device while the AI is tasked with cutting. These systems relieve humans of part of their task, either because the AI can do it better or because the human is unable to do it at all.

#### **5.2.2 Supervision—Do Actors Correct Each Other?**

A shared control system's design embeds assumptions about the supervisory relationships among actors. For example, the SAE's levels of automation assume that human drivers of semi-autonomous vehicles, who until recently have been solely responsible for all primary driving tasks, are more capable drivers than the AI. They are therefore responsible for supervising the AI agent's control and overriding it when necessary [228]. In this way, one actor intervenes to prevent the other actor's

mistakes. For example, a driver using Navigate on Autopilot (D7) may not know exactly what the AI is doing, only that they don't like it, and choose to take back control. Many systems assume a one-way supervisory relationship, with either the human supervising the AI (G8, D2, D6-7, D9, C7-8, R6, S4) or the AI supervising the human (G5-7, D5, C3-5, C12, M1-9, R1, R4-5, S3, S7-8). Some systems make no such assumptions (G1-4, D1, D8, C1-2, C9-11, R3, R9, S1-2, S5-6), while others afford mutual supervision (D3-4, C6, R2, R7-8) where AI and human each supervise the other. Our Supervision dimension captures the supervisory responsibilities of actors according to these four categories: *Unsupervised*, *By AI*, *By Human*, and *Mutual*.

#### **Unsupervised: Neither actor supervises the other**

When both actors are *Unsupervised*, neither actor controls the system to prevent the other from making mistakes. Their intentions may still conflict, such as when a smart power wheelchair steers to the right to avoid an obstacles while the human steers to the left, but not because one believes the other to be in error. For example, EMS Air Guitar (C2) helps humans to strum an imaginary guitar along with music. The system has no sensors, and is therefore unable to react to anything, so it stimulates the human's arm muscles without supervising their movements.

#### **By AI: The AI supervises the human**

Many AI agents are designed to monitor the human and take action when the human errs. For example, every mobility assistance system we surveyed (M1-9) uses AI to supervise the human and amend or override their control to prevent collisions. This form of supervision is applicable when the AI is able

to detect that the human has erred; for example, the craniotomy tool Cranioscan (S3) tracks a reference path across a patient's skull and steers the drill towards the path when it deviates.

#### **By Human: *The human supervises the AI***

Conversely, shared control systems can be designed such that the human user supervises the AI. For example, Supervisory Steady-Hand (S4) performs microinjections fully autonomously, but requires a surgeon to supervise its performance and verify that it succeeded.

#### **Mutual: *Both actors supervise the other***

Finally, shared control systems can be designed such that both actors supervise each other. For example, the Mavic Air 2 supervises the human to help with collision avoidance during manual operation, and the human needs to supervise the AI when it flies autonomously. In lane assistance driving systems (D3-4), the AI supervises the human's steering, taking over when the car drifts out of lane; the human in turn supervises this correction, and can override it by applying extra force to the steering wheel.

### **5.2.3 Influence—*Do Actors Attend to and Influence Each Other?***

To cooperate effectively, both human and AI may need to monitor what the other is doing to determine how they should control the system together. In some systems, the AI's control signal is a function of the human's. For example, the racing game AI of Cechanowicz et al. (G7) only assists the player when it detects that they are steering. In other systems, the human responds to the AI's control signal, which they

perceive via force feedback (D1, C3-6), EMS (C1-2), or a visual display (C7). For example, Haptic Intelligentsia (C4) helps users to construct sculptures out of glue by guiding their control of a hot glue gun. When the user moves the gun outside a predefined 3D volume, known to the AI but not to the user, a haptic device pushes the gun back towards the volume's surface. This informs the user of the volume's location and helps them to decide where to put more glue. In this way, users of these systems are made aware of the AI's commands and may infer and react to its intentions. When both the human and AI each respond to the other's control signal, they can communicate with each other and negotiate how the system is controlled. We term the way in which human and AI interpret and respond to the other's control signal Influence, which we have observed as being *Independent*, *Interpreted*, *Guided*, and *Codependent*.

#### **Independent: *Neither actor influences the other***

In many systems, there is no obvious benefit to actors influencing each others' control. For example, in the *Space Invaders* clone *Alienated* (G2) the player controls the avatar's movement while the AI controls its shooting. Since movement does not affect shooting, the AI's decision to fire is determined by the game's state, rather than the player's actions. Neither actor is directly aware of the other's commands.

#### **Interpreted: *The AI's actions are influenced by the human's actions***

When the AI interprets the human's control signal, it may be able to infer and help the human achieve their goals. For example, *Gekku Aim* (G6) is a 2D racing game designed for children with cerebral palsy who have difficulty aiming at targets. When the player aims near an enemy and fires a shot, the

AI steps in at the last moment and aims directly at the enemy closest to where the player was pointing. In this way, the AI may interpret users' control signals to infer their intentions and then take action to help the human realize them.

#### **Guided: *The human's actions are influenced by the AI's actions***

Several systems use EMS or force feedback to guide the human's control. For example, Kianzad et al. created a physically assisted sketching system (C6) that pushes the user's pen away from bounding lines they have drawn on paper, helping them to stay within a predefined region. This form of Influence can guide the human's control not only by physically pushing or pulling the system's hardware but also by improving the human's awareness. For example, the force-amplifying surgical robot created by Payne et al. (S6) improves kinaesthetic feedback from the tip of the surgeon's instruments, which is otherwise not present in minimally invasive surgery. This enables surgeons to sense what the AI senses and use their heightened awareness to operate more precisely.

#### **Codependent: *Both actors' actions are influenced by each other's actions***

When both actors are aware of the other's control signal, they may infer the other's intentions to cooperate better. If both actors control the same parts of the system's interface, they may engage in a negotiation to determine their form of control. For example, Shilkrot et al. created a digital airbrush (C12) that uses force feedback on its trigger to help users paint a reference image. When the human pushes down on the trigger to paint over an already painted

area, the AI pushes back. Therefore, both actors influence each other by communicating their intentions through the trigger. This specific form of *Codependent* shared control, in which control is negotiated using physical forces, has been called *haptic shared control* by Abbink et al. [3]. It is especially applicable to systems where actors need to maintain awareness of the other's control in real time, such as semi-autonomous vehicles (D3-4, D9) and surgical robots (S1-5).

#### 5.2.4 Mediation—*How are Actors' Commands Unified?*

Sometimes, the actions that the human and the AI agent perform may conflict. For example, the image classifier of Seno et al. (C7) may incorrectly label an object as a car while the human correctly labels it as a horse. The Mediation dimension captures how such conflicts are resolved. More technically, this dimension describes how actors' commands are used to control the system.

##### **Combined:** *Actors' commands are combined to construct the shared control signal*

A system's mediation of actors' control signals is said to be *Combined* if the shared control signal is always a combination of both actors' control signals. For example, a human driver may turn their steering wheel to change lanes while their car's lane-keeping AI turns the wheel in the opposite direction, negating the human's action because they did not signal. In this example, actors' commands are *Combined* physically to create a new command. Both actors contribute to this new command and may control the system more forcefully, turning the wheel further in this case, to override the other actor's contribution. This approach is used by several lane-keeping assistants (D1-4).

**Selected: One actor's command is selected as the shared control signal**

Alternatively, a system might select one actor's control signal as the preferred control signal, in which case its mediation is said to be *Selected*. These systems continuously check if the human or the AI should be solely in control. For example, Sentouh et al. created a lane-keeping assistant (D5) that monitors the human driver's commands and selectively ignores them when they might violate stabilization and lane-keeping constraints. Instead of combining both actors' commands, this assistant decides which actor should steer and *selects* their command to control the vehicle. In some *Selected* systems, actors' commands are both *Combined* and *Selected* at different times. For example, G8 combines the player's shooting commands with the AI's aiming commands, but selects the player's aiming commands when they take control of that input. This type of Mediation has important consequences for the human-AI interactions systems afford. *Selected* systems enable actors to remove the other from the control loop, which means that one actor can act autonomously and without interference.

### 5.3 Shared Control Design Patterns

As we have shown, our dimension space describes human-AI interactions in shared control along four dimensions: *AI Role*, *Supervision*, *Influence*, and *Mediation*. The dimension space enables designers to classify systems and discover patterns in the designs used within a problem domain. Our dimension space can help designers to identify common assumptions about actors' roles, responsibilities, competencies, and ways of cooperating and then imagine how these interactions might otherwise

be structured. In this section, we apply the dimension space to the systems from which it was derived to demonstrate how it can elucidate common design patterns. We then apply it to our entire corpus to identify larger design patterns used across domains.

### 5.3.1 Domain-Specific Design Patterns

Designers can classify a shared control system by assigning it to one category in each dimension in the dimension space. When multiple systems from the same problem domain are classified, patterns emerge, providing insight into to the types of interactions they afford. For example, human users of mobility assistance systems are typically supervised *by the AI* who *supports* them in avoiding collisions and *interprets* their commands to infer their intentions. However, they may either *combine* or *select* actors' control signals. Therefore, mobility assistance devices exhibit a *Supportive*, *By AI*, and *Interpreted* design pattern, defined along three dimensions (leaving the category in the fourth dimension open). We have called this pattern *AI supervisory control* in mobility assistance devices, but in games it is already known as *player balancing*. Therefore, similarities in the patterns used in multiple domains may reveal correspondences in the interactions systems afford and users' experiences of them. Design insights discovered in one domain may be directly applicable in designing systems for another.

Of course, identifying design patterns is only the first step. Once we know how a set of design choices overcomes domain specific problems, we can better understand how similar designs might overcome other problems. We can describe what

about a pattern makes it effective and imagine how the strengths of multiple patterns might be combined to provide novel forms of shared control. In this section, we demonstrate how our dimension space enables designers to classify, compare, and understand shared control solutions to similar problems by identifying common design patterns within each of the six domains we surveyed. We present eight patterns drawn from dimension space plots of our 55 surveyed systems, shown in Figures 5.2-5.7.

### 5.3.1.1 Digital Games

Player balancing enables weaker players to compete with stronger opponents. For example, in *Gekku Aim* (G2) players shoot projectiles at each other in a 2D play area and aim their shots using a gamepad's analog stick. The game assists players with deficits in manual dexterity by aiming directly at an opponent when their aim is misaligned but close. This makes the game easier to play for players who experience difficulty. As shown in Figure 5.2, *Player Balancing* is a *Supportive*, *By AI*, and *Interpreted* design pattern. The AI's role is to *support* the player's activities. To do so, the AI *supervises* the player and *interprets* their aiming actions, modifying this control signal to provide the player with improved aim. The pattern is agnostic with respect to Mediation strategy. Figure 5.2 shows two games whose aim assistance algorithms follow this pattern.

A second design pattern for games, *Partial Automation*, is also shown in Figure 5.2. This pattern delegates control of inaccessible game inputs to an AI copilot. For example, in *Zac - O Esquilo*, a one-switch clone of *Frogger*, the player presses a button when they want their avatar to move [203]. The copilot then selects

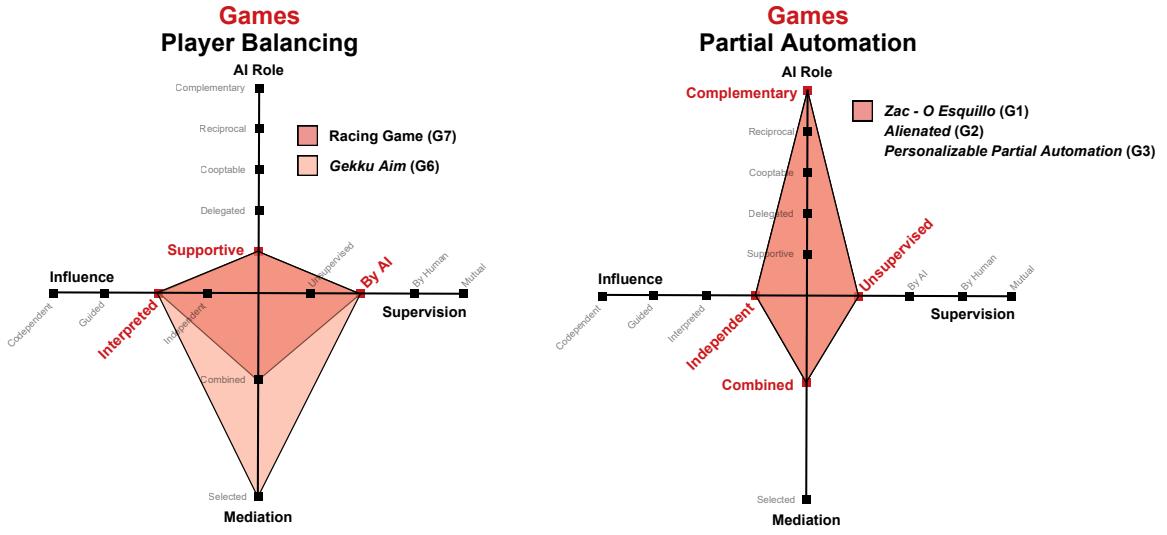


Figure 5.2: The two design patterns identified in digital games. Although both are forms of partial automation, we use the term *player balancing* to distinguish partial automation that amends players control to assist with shared tasks such as aiming and steering.

the movement direction it deems most appropriate and the avatar moves in that direction. While the player controls both *when* and *where* their avatar moves in the original game, *Zac - O Esquilo* partitions the player's task by asking them only to choose *when* to move. Games like *Zac - O Esquilo* use partial automation to make games nearly universally accessible by adhering to a *Unsupervised*, *Complementary*, *Independent*, and *Combined* design pattern. Actors perform different tasks simultaneously and therefore do not directly supervise each other (*Unsupervised Supervision*) or influence each other (*Independent Influence*). The AI's Role is to complement the player's actions. The player and AI's inputs do not conflict and are therefore Combined as a *Mediation* strategy.

As we will demonstrate in Chapter 6, partial automation can enable players with radically different physical abilities to play the same games, by broadening their accessibility to players they were not designed for. However, as we will demonstrate in Chapter 7, players may experience *automation confusion* [52] if they do not understand what the copilot is doing, because it performs a *Complementary* role, or become frustrated when it does things they do not like, because their control signals are *Independent*.

### 5.3.1.2 Semi-Autonomous Driving

Early semi-autonomous driving systems, such as lane-keeping assistants, sought to make simple driving tasks easier. These systems represent a transition from *low-to-high* automation, introducing automation into manual driving. These systems follow a *Supportive*, *Combined*, and *Guided* pattern that enables the AI to perform operational level driving tasks at the same time as the human (Figure 5.3). The AI *supports* drivers as they both steer to stay in the lane. AI in low-to-high automation uses force feedback to *guide* the driver's steering, which makes lane keeping easier by *combining* steering forces from both actors. The pattern is agnostic as to Supervision strategy.

As more automated features have been integrated into cars, designers have imagined novel interaction metaphors and design philosophies for semi-autonomous driving. Consequently, a *high-to-low* automation pattern has begun to emerge, introducing manual control into automated driving. For example, the *H-Metaphor* suggests that semi-autonomous vehicles should cooperate with drivers in the same way horses cooperate with riders, in that they can act autonomously or allow the

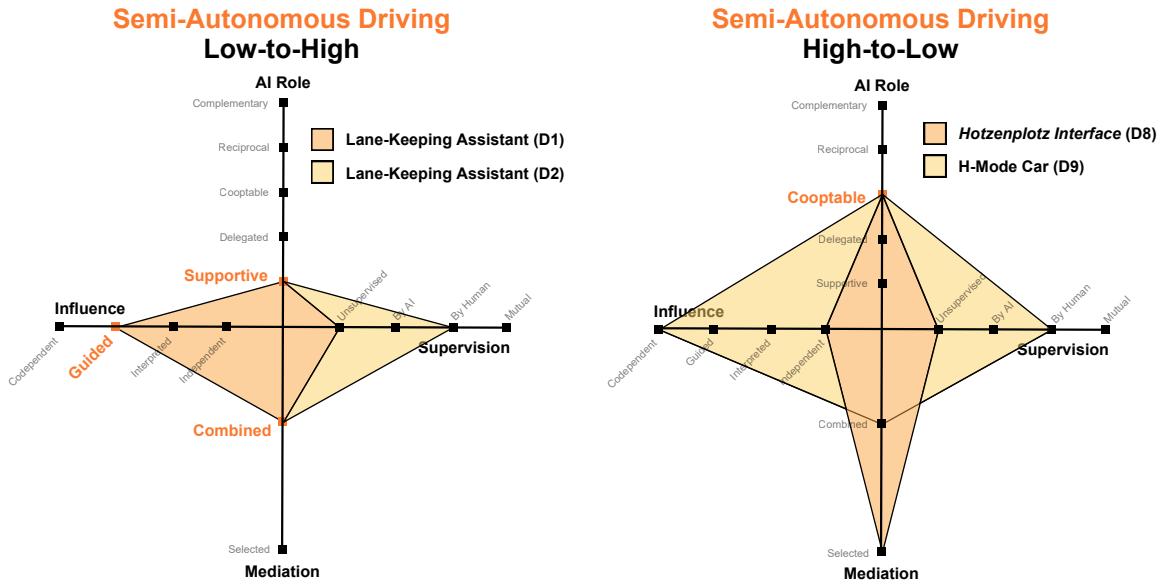


Figure 5.3: The two design patterns identified in semi-autonomous driving.

human to take control [288; 91]. Otherwise, designers may see automating driving tasks as “*an amputation*” [163] of the driver’s task and seek to provide them with playful ways to engage in driving [93; 318]. High-to-low systems put the human in a position of greater authority and put the AI in a *Cooptable* role. Systems adopting this approach differ in their choices of Influence, Supervision, and Mediation approaches, indicating that the design of high-to-low systems is still under active exploration.

### 5.3.1.3 Creativity Support

Several of the creativity support systems we surveyed were designed to help novices sketch (C3, C6, C9-10) or play a musical instrument (C1-2). They use EMS and force feedback to guide creators’ control of their tools, enabling them to perform skilled actions that may be too difficult without assistance. These systems represent

a form of shared control that we have called *guided performance* (Figure 5.4), in which the human is *Guided* by the AI which performs a *Supportive* role while actors' control signals are *Combined*. Example systems use different forms of Supervision.

Since guided performance systems are designed for novice users, the AI's guidance is intended to help them overcome technical barriers to creative endeavors. For example, sketching assistants help humans to draw straight lines (C3) and stay within them (C6), both of which are tasks that the human may learn to do on their own with more experience. Therefore, this pattern helps novices to get acquainted with their craft by removing technical burdens and enabling creators to create freely.

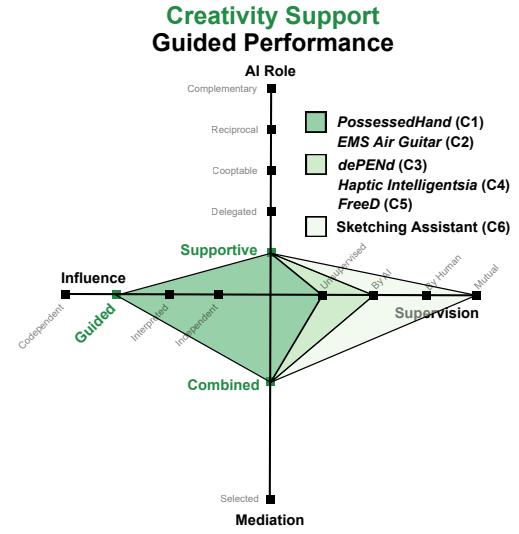


Figure 5.4: The design pattern identified in creativity support.

#### 5.3.1.4 Mobility Assistance

Mobility assistance is one of the more popular applications of shared control and was one of the more homogeneous domains we surveyed. Since the user is expected to have difficulty controlling these devices, the AI is put in the role of supervisor and is often permitted to override the user's control when it sees fit. These systems interpret the human's control signal to infer their intentions and may ignore their commands if executing them is dangerous. Therefore, the human's control signal is

*Interpreted* as they are supervised *By AI*, which plays a *Supportive* role. These systems have the AI supervise the human, sometimes preventing them from controlling the device at all, and therefore adhere to a design pattern that we have called *AI supervisory control* (Figure 5.5). The pattern does not specify a Mediation strategy.

Unfortunately, little has been reported about users' experiences of interacting with these devices, since many evaluations have not included disabled users (e.g. [81; 67; 188; 189; 314; 279]) and authors are typically more focused on the technical challenges of avoiding collisions. However, what is known indicates that these devices can help disabled users to navigate more safely [80; 99]. If interacting with these systems is anything like interacting with games that use player balancing, which exhibit the same design pattern, then users may find the AI's intervention helpful but potentially intrusive.

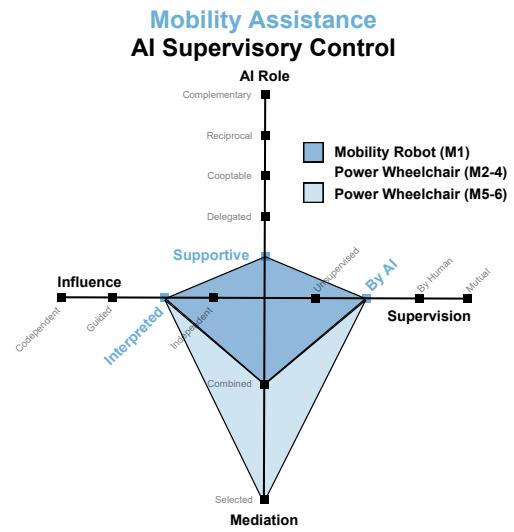


Figure 5.5: The design pattern identified in mobility assistance.

### 5.3.1.5 Telerobotics

As explained by Jiang & Odom, the history of human-robot team design marks an ideological shift from seeing robots as subordinate tools to seeing them as equal partners [148]. Regardless of whether it is posing a robot arm to make manipulation easier or stabilizing a drone to counteract signal interference, teleoperated

robots are being designed to overcome technical and human factors problems by leveraging their unique strengths.

For example, Dragan & Srinivasa created a telemanipulation robot to assist remote operators in picking up objects [71; 70; 72]. They point out that human operators have a better understanding of the task than the robot (e.g., knowing that caution is needed around breakable objects), although they have difficulty controlling the robot precisely and can become fatigued over time.

In contrast, an AI partner is tireless and can control a robot perfectly precisely. In order to account for the deficiencies of one actor by leveraging the strengths of the other, the AI predicts which object the user is trying to grab and assists them in doing so safely. However, unlike *AI supervisory control*, these systems do not necessarily have the AI supervise the human, since the human is assumed to have greater authority in some situations. Therefore, teleoperated robot systems typically adhere to a *Supportive* and *Combined* pattern, which we have called *daemon assistant* (Figure 5.6), in which the AI works in the background to *support* the human by *combining* their commands.

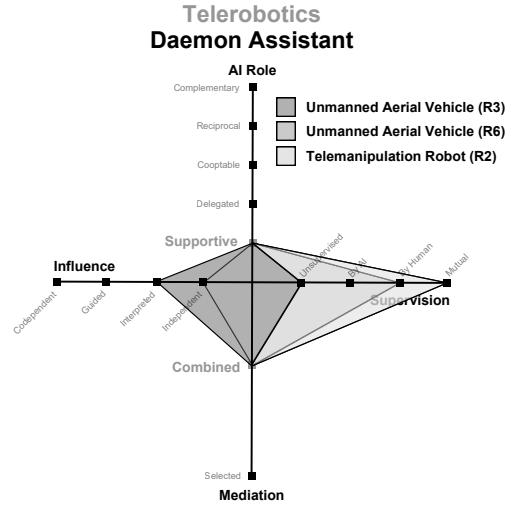


Figure 5.6: The design pattern identified in telerobotics.

### 5.3.1.6 Surgery

Surgical robots help surgeons to perform delicate and precise operations by providing haptic guidance and filtering tremulous movement. Because the surgeon is the designated expert in the operating room, the AI agents in these systems are seldom capable of performing tasks on their own, although some examples can be found (S4-5). These systems also vary with regards to actors' supervisory responsibilities. The surgeon is authoritative, but not infallible, so some systems let the AI supervise (S3, S7-8) while others are designed for neither actor to supervise the other (S1-2, S5-6).

Surgical robots therefore adhere to a *Combined* and *Codependent* design pattern, which we have called *negotiated control* (Figure 5.7). Since actors sense the other's actions via the forces they exert on the surgeon's implements, actors negotiate how the system should be controlled. Each responds to the other's movements, which move the surgeon's tools when *combined*, so their action is *codependent*. If the AI believes the surgeon to be in error, the surgeon may discover their mistake when the AI counteracts their movement using force feedback. This approach can provide surgeons with superhuman precision with their implements and superhuman awareness of their workspace.

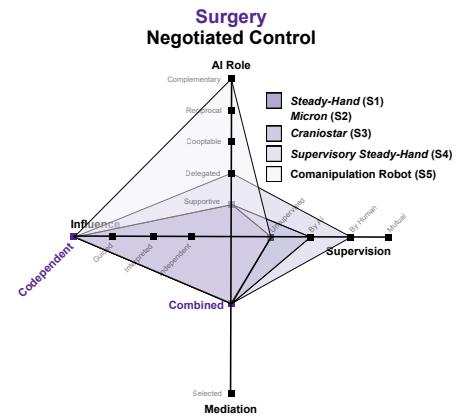


Figure 5.7: The design pattern identified in surgery.

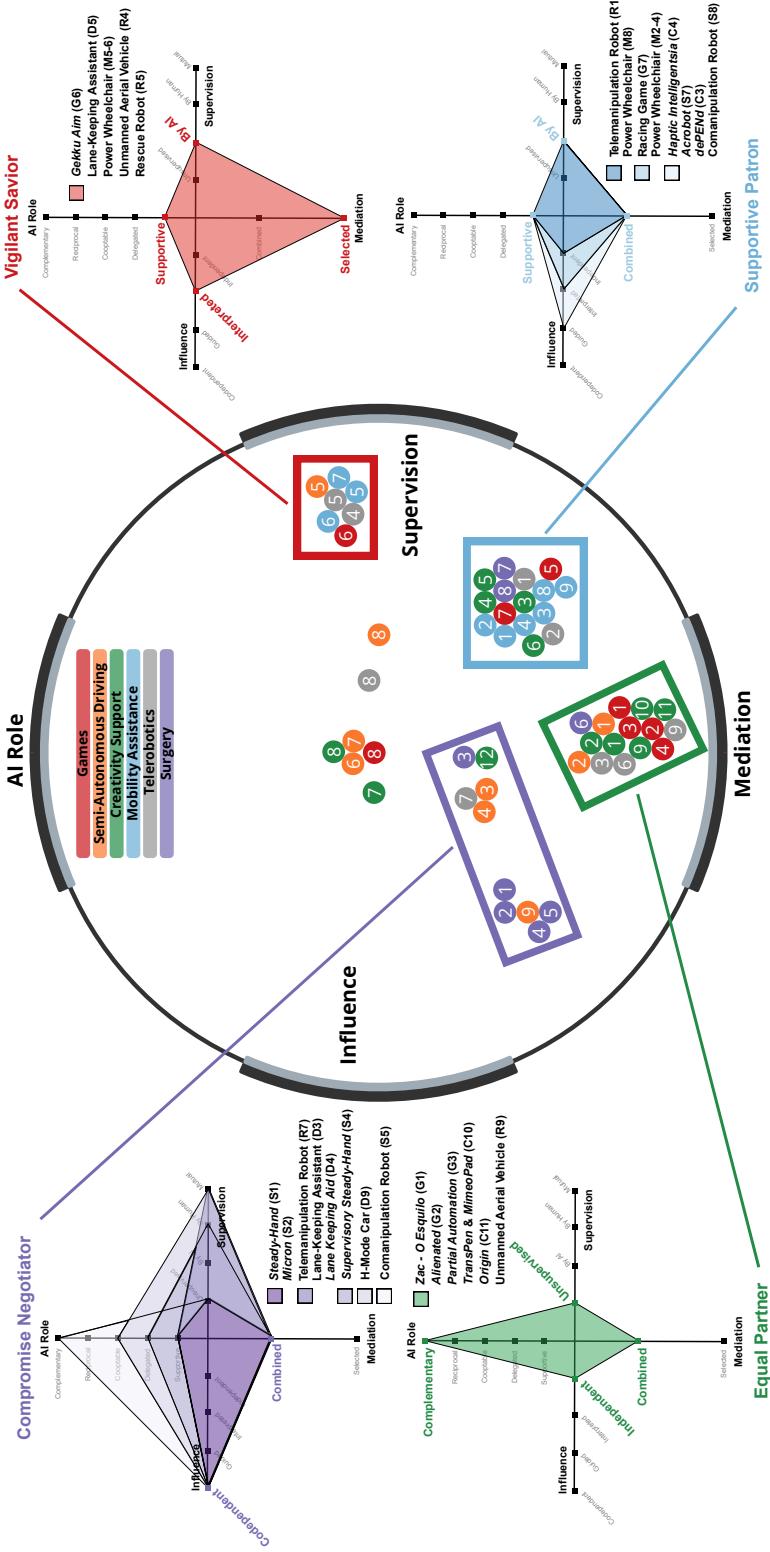


Figure 5.8: The *data view* RadViz plot generated by the *Subspaces Explorer* tool and higher-level design patterns used across multiple domains. In the data view in the center, each of the systems in our corpus are represented as circles colored according to their domains and plotted according to their similarities with other systems as well as the axes along which they are similar. For example, the cluster in the top right corner use a distinctive combination of Influence and Mediation, while the cluster at the bottom places the AI in a unique AI Role. Systems found in the center of the plot did not exhibit a cohesive pattern, so this cluster was not selected. Selected patterns are enclosed in colored rectangles that each correspond to a diagram of the same color on the right and left.

### 5.3.2 Design Patterns Across Domains

Having shown how human-AI interactions are structured in various domains using shared control, we now apply our dimension space to our entire corpus to determine which designs are used across domains. In so doing, we elucidate commonalities in the design of these systems and describe the forms of shared control they provide. These patterns tell us not only how shared control is used across multiple domains but also what problems these domains have in common. Because designers working in the domains we surveyed may be unaware of how shared control is used by others, these patterns represent more general forms of shared control that have been discovered independently multiple times. The broader perspective we take in this section enables us to identify the most popular design patterns and understand how common design choices overcome the most pervasive problems in controlling interactive systems.

To identify these patterns, we used Artur & Minghim's *Subspaces Explorer* system, shown in Figure 5.8, which uses correlation analyses to embed and cluster data in an alternative RadViz space [12]. The color coding and spatial arrangement of systems plotted this way are explained in the figure caption. We then selected homogeneous clusters and combined them to create patterns, such that each had a similar number of systems that adhere to it (i.e., 7 to 16). The cluster in the center of the RadViz plot in Figure 5.8 contains the outliers that did not belong to a more cohesive pattern, so this cluster was not selected. In this section, we describe the four higher-level design patterns we identified.

### 5.3.2.1 Vigilant Savior

The *vigilant savior* pattern (Figure 5.9) has the AI step in to override the user's control in dangerous situations. The human is supervised *By AI*, which performs a *Supportive* role. When the AI detects that the human is in need of assistance, the human's control signal is *Interpreted* by the AI whose control signal is *Selected* as the shared control signal. For example, Jiang et al. created a UAV (R4) that predicts whether the human's command would put it into an unsafe state and overrides their control if it would.

This pattern is a more specific form of the player balancing and AI supervisory control patterns described in the last section. However, this pattern has also been used in semi-autonomous vehicles for lane-keeping (D5) and in teleoperated robots to take over control when the human performs poorly (R5). In this way, vigilant savior systems assume a deficiency in the human's capabilities and override the human's control to prevent them from making mistakes, saving the day.

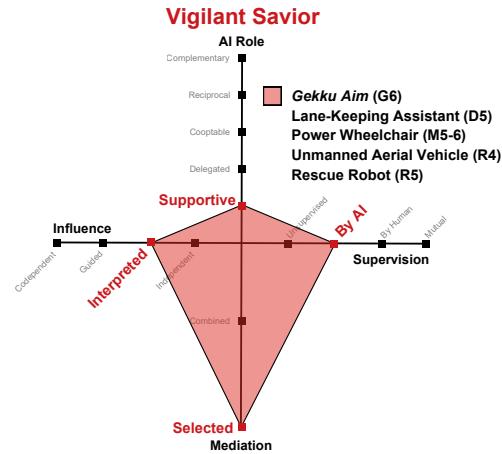


Figure 5.9: The *vigilant savior* design pattern.

### 5.3.2.2 Supportive Patron

Like the vigilant savior, the *supportive patron* pattern (Figure 5.10) has the AI supervise the human and support them with their tasks as needed. However, this

pattern uses *Combined* control and the AI does not necessarily interpret to the human's commands. Rather, the AI is *Supportive* of the human, who is supervised *By AI*, and their control signals are *Combined* to perform the same tasks at the same time.

The pattern is used across a range of Influence styles. For example, Deng et al. created a smart power wheelchair (M3) that refines the human's control by blending it with the control signal of an autonomous path planner that moves away from obstacles. Should the user command their wheelchair to move towards an obstacle, blending nullifies the user's command and prevents the collision. When the wheelchair gets too close to an obstacle, the planner adjusts the user's course to give them the space they need to operate the wheelchair safely. This is similar to how the comanipulation robot of Gurnel et al. (S8) pushes and rotates the surgeon's implements to guide them towards a desired position and orientation. Instead of stepping in to replace the human's control signal, this pattern's AI uses shared control to make the human's tasks easier by performing the tasks at the same time.

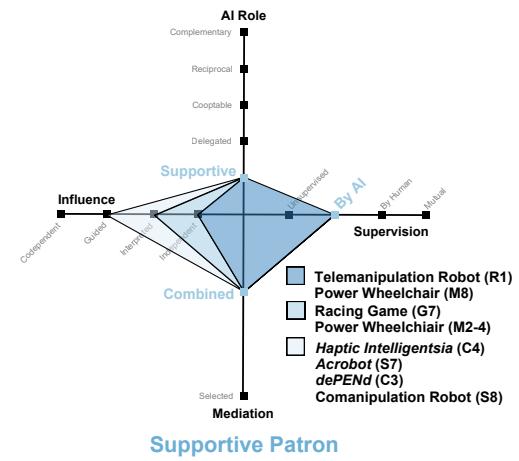


Figure 5.10: The *supportive patron* design pattern.

### 5.3.2.3 Compromise Negotiator

Many AI agents both interpret the human's control signal and use their own to guide the human. These provide a form of *Codependent* and *Combined* shared control that

Abbink et al. have previously called haptic shared control [3]. For example, the haptic shared control vehicle presented by Johns et al. (D3) enables both actors to steer and communicate their intentions using the steering wheel (Codependent Influence). Both the human and AI negotiate the shared control signal by simultaneously applying forces to the system's physical interface (Combined Mediation). However, as described in Section 5.3.1.3, the pattern encompasses other ways in which AI's control signal can be communicated to the human. The *compromise negotiator* pattern (Figure 5.11) encompasses these haptic shared control systems, but is instead defined in terms of the Mediation and Influence of actor's control signals. The AI interprets and guides the human's control to negotiate a shared control signal that both actors find agreeable.

As seen in Figure 5.8, numerous systems with differing AI Role and Supervision styles follow the Compromise Negotiator pattern. For example, the digital airbrush of Shilkrot et al. (C12) selects which color to paint and supervises the user, negotiating how much paint to apply. The H-Mode car (D9) is supervised by the human, who coopts the AI's tasks, negotiating how quickly the vehicle accelerates and turns. Many surgical robots (i.e., S1-5) use this pattern to negotiate how the surgeon handles their implements.

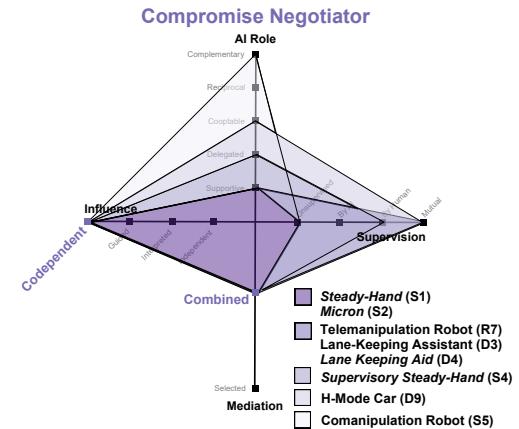


Figure 5.11: The *compromise negotiator* design pattern.

### 5.3.2.4 Equal Partner

When AI agents share control to perform their own tasks that are *Complementary* to the human's, they can extend the human's capabilities by performing tasks that the human cannot, or partition the human's tasks to make the human's job easier. For example, a mobility assistance robot could brake to catch the human if they fall (M9) or an endomicroscopy robot could rotate a scanner while the human translates it across a sample (S5).

Since the AI performs tasks that the human does not, their control signals are often *Independent*, *Combined* with equal control authority, and *Unsupervised*. This form of shared control frames the AI as an *equal partner* (Figure 5.12) simultaneously performs its own tasks. It has been used in each of the domains we surveyed, save for semi-autonomous driving.

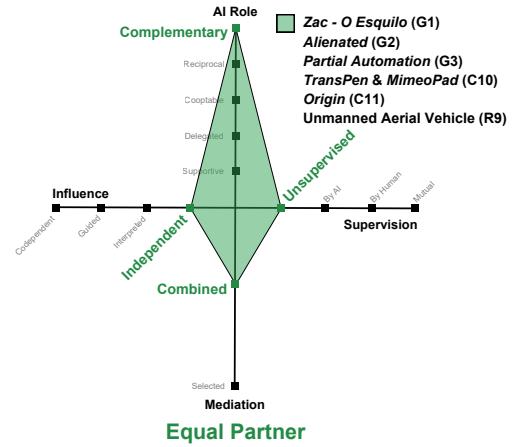


Figure 5.12: The *equal partner* design pattern.

## 5.4 Generative Shared Control Design

Human-AI shared control can personalize the control of systems to the abilities of users, by leveraging the abilities of one actor to extend the abilities of the other. It unifies human and artificial intelligence, enabling humans to play and create without barriers, get around more safely, and work more effectively. The systems we surveyed demonstrate how shared control can improve human users' interactions

with computers in tasks as diverse as surgery and sketching. We turn our attention now to the future of shared control and demonstrate how our dimension space might help designers to explore the design choices available to them.

We have shown how our dimension space supports analysis of shared control systems at different scales. Applying the dimension space to individual systems (as in Figure 5.1c) enables designers to understand and compare the specific human-AI interactions they afford. By using the dimension space to classify a more diverse sample of systems (as in Figure 5.8), designers can gain insights into how broadly applicable design patterns can address problems encountered in their own domain. However, there is still one further use case for our dimension space that we demonstrate in this section. Once known solutions are classified, designers can use the dimension space to imagine how other types of Supervision, AI Role, Mediation, and Influence might shape the interactions that systems afford. In this section, we provide examples of designs generated with aid of the dimension space.

We propose novel designs for shared control systems in three of the domains we surveyed. Our proposals, depicted in Figures 5.13-5.15, are largely speculative and there may be concrete human factors or technical problems that preclude their creation. Section 5.4.1 describes how the Equal Partner pattern, modified to provide *Interpreted* Influence, might make driving more accessible to persons with disabilities. Section 5.4.2 describes how *Cooptable* AI, similar to the *High-to-Low* pattern used in semi-autonomous driving, might overcome an emerging problem in the performing arts. Section 5.4.3 describes how a *Guided* telemanipulation system might help users to understand the AI's assistance.

### 5.4.1 Semi-Autonomous Driving

Designers of semi-autonomous vehicles tend to approach the problem from one of two perspectives: removing manual control from driving (low-to-high automation) or introducing manual control into autonomous driving (high-to-low automation). However, both assume a one-dimensional shift in control authority between human and AI. These approaches are appropriate for persons without disabilities, but the advent of autonomous vehicles has also sparked discussion regarding how they might better serve persons with disabilities. Brewer & Kameswaran asked persons with vision impairments how automation might enable them to drive and discovered that they wanted autonomous vehicles to present drivers with a “spectrum of desired control” [33]. They wanted to personalize their control of the vehicle based on their abilities, rather than designers’ expectations of their abilities.

We did not encounter any systems that partition driving tasks to make driving more accessible to drivers with disabilities. For example, many people with spinal cord injury find aspects of driving inaccessible. They may customize their vehicles with hand controls instead of pedals or pedal controls instead of a steering wheel, but for some there may be no configuration of assistive technologies that

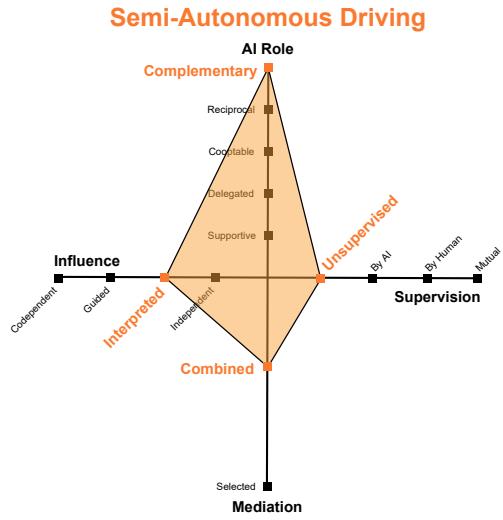


Figure 5.13: The Kiviat diagram representing our novel semi-autonomous driving system.

make all tasks safe and accessible [125]. Using shared control, designers could automate different aspects of the driver's task separately and drivers with spinal cord injury could perform whichever tasks are accessible to them. The AI could perform an entirely *Complementary* role, filling in for the driver by performing tasks that they cannot. Using *Interpreted Influence*, the AI could look for clues in the driver's commands that indicate what the driver wants to do next (Figure 5.13). For example, the AI could change lanes when the driver accelerates towards slower vehicles ahead. In accordance with the equal partner pattern, in which the AI performs a *Complementary* role, it may be appropriate for neither actor to supervise. Just as partial automation controls inaccessible inputs to make games more accessible, AI performing a *Complementary* role could make driving more accessible to drivers with motor disabilities.

### 5.4.2 Creativity Support

*The Under Presents: Tempest* [55] is a production of William Shakespeare's *The Tempest* staged live in virtual reality (VR). A lone actor, of the thespian variety in this case, plays multiple roles throughout the performance by controlling virtual character models, although they can only play one part at a time. Were the performance to require that multiple parts be played simultaneously, then multiple actors would be

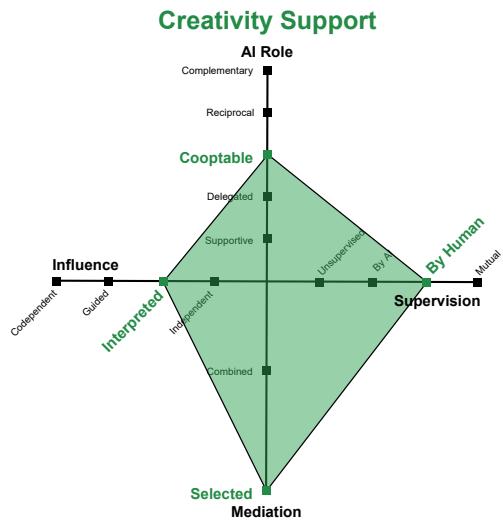


Figure 5.14: The Kiviat diagram representing our novel creativity support system.

needed. Instead, productions could be scaled up dramatically by sharing control of each virtual character with an AI agent that performs scripted behaviours autonomously. The cast could supervise these agents and selectively coopt their roles. For example, should an audience member try interacting with a virtual character without any scripted dialogue, a cast member might take control of this character and improvise. While the character is being controlled, the AI could monitor the cast member's movement, to infer which actions they are doing, and continue doing those actions when control is relinquished. In this way, these *Cooptable* agents are supervised *By Human* and the human actor's commands are *Interpreted* when they are *Selected* to control the character (Figure 5.14).

### 5.4.3 Telerobotics

Telerobots are operated at a distance, so human users may have lower awareness of the robots' environment than the robot itself. For example, operators of a robot arm may be unsure whether the arm's gripper has successfully picked up an object using visual feedback alone [238]. For this reason, haptic feedback has been used in telemanipulation systems to improve operators' telepresence. However, we encountered no telerobotic systems that use haptic feedback

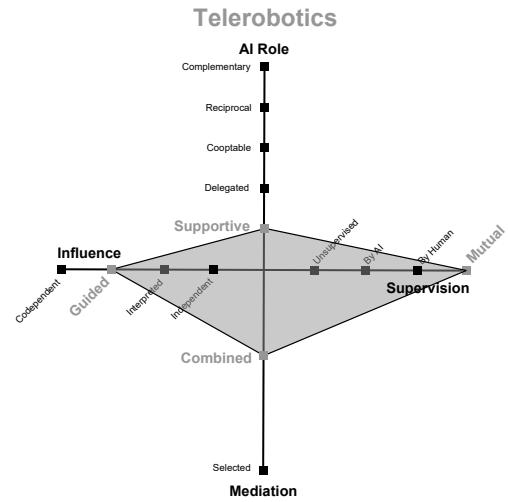


Figure 5.15: The Kiviat diagram representing our novel telerobotics system.

to share control. This is unfortunate because controlling these robots may be difficult [248; 71; 72] and unsafe in some environments [238]. Sharing control may help operators to avoid making mistakes (e.g., R1-2), but without adequate awareness of how the AI has amended their commands, operators may find control confusing. Instead, telerobots could be *Supportive* of operators and inform them of the AI’s commands, such that their supervision of each other is *Mutual* and *Guided* using force feedback (Figure 5.15). For example, systems in which actors’ joystick commands are *Combined* could position the user’s joystick to reflect the *Combined* command the system received. Users could sense that their command is dangerous, not because the robot’s sensors indicate they might hit an object but because the AI has already prevented it. This approach may improve operators’ awareness of the robot’s environment, actions, and intentions while also providing the improved task performance afforded by haptic feedback.

## 5.5 Summary

Throughout this chapter, we have described many systems that use shared control to solve similar problems in significantly different ways. We have also shown how designing novel uses of shared control benefits from our existing design knowledge and exposes new problems that we know little about. It has only been in the last decade that shared control has emerged as a way of controlling interactive systems beyond the relatively narrow design space of robots. These recent uses of shared control in games and creativity support suggest a future in which shared control is used to improve all sorts of human activities. As shown by the speculative examples presented in this section, the dimension space presented in this chapter provides

designers with the language to express how shared control might overcome new problems. We are now equipped to go beyond the domains where shared control has proven useful, using this dimension space to guide our exploration.

In the next chapter, we will apply our dimension space to a previously unexplored issue in games accessibility. As explained in Chapter 2, people with spinal cord injury need to undergo lifelong physical rehabilitation to support their independence and health. A spinal cord injury can cause a range of motor impairments, from little or no impairment to complete paralysis, so the physical abilities of people with spinal cord injury as a population are diverse. As explained in Chapter 3, this severely restricts the accessibility of exergames to people with spinal cord injury. Exergames can motivate patients to do exercises that they consider boring and thereby improve patients' experiences of doing these exercises, but only if they are able to play. Interface adaptations can extend a game's accessibility to players who are able to use them, but many individual adaptations may be needed to extend access to the entire population. However, we saw in this chapter how shared control systems that adhere to the *equal partner* pattern can excise inaccessible tasks from the human's control and perform these tasks autonomously. To substantiate our claim that partial automation can extend the accessibility afforded by other approaches to games accessibility, in the next chapter we will propose a personalizable form of partial automation that combines interface adaptation with the *equal partner* pattern and demonstrate its capacity to broaden a universally designed game's accessibility to players with different abilities due to spinal cord injury.

# CHAPTER 6

---

## Personalizable Partial Automation

---

PARTIAL automation is a form of human-AI shared control that has been used in a handful of commercial and hobbyist games. It may make games more accessible to players who have difficulty providing input, but accessibility benefits of partial automation have yet to be studied rigorously or systematically. In Chapter 2, we discussed the accessibility barriers faced by players with motor impairments and explained how the interaction demands imposed by gaming disable many. As described in Chapter 3, game designers can employ universal design, interface adaptation, player balancing, or a combination of these approaches to broaden a game’s accessibility to players who were previously unable to play. However, these approaches cannot relieve players of all of a game’s interaction demands that might disable them. Players still need to perform all of the actions involved

in playing the game; this would constitute the exercise in an exergame, long sequences of button presses in a fighting game, or simultaneously tilting two analog sticks in a third-person action game. If a player is unable to do these things, then they may be unable to play and thus may require a different form of assistance. Hence, partial automation can relieve players of disabling interaction demands by sharing control of the game.

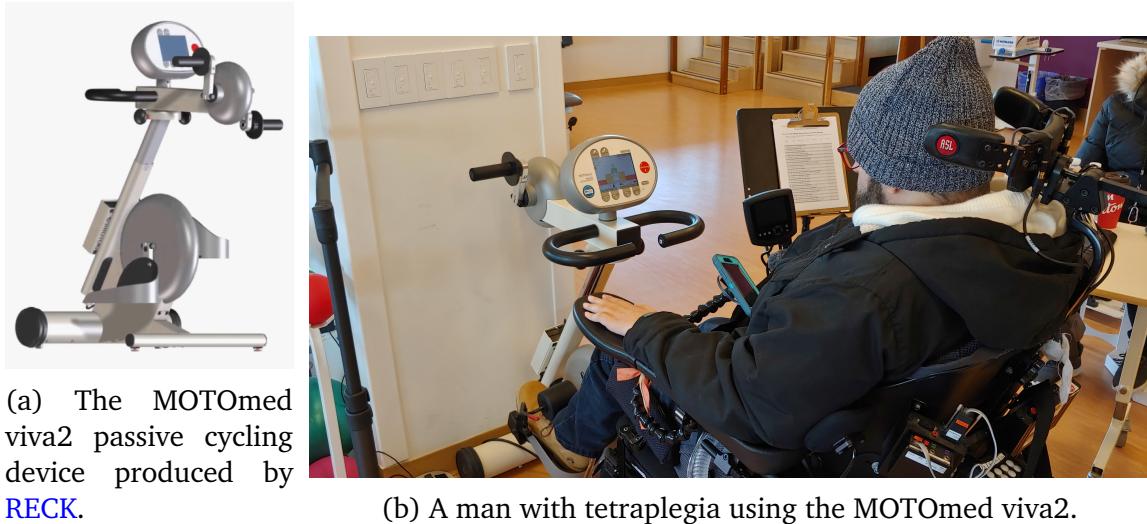
In this chapter, we put partial automation to the test and demonstrate its capacity to extend the accessibility afforded by other games accessibility techniques. As explained in Chapter 5, the *equal partner* shared control design pattern can excise disabling tasks from users' control and perform a complementary role, in which the copilot does the tasks that the human cannot. To extend the accessibility of other approaches, we created a novel form of partial automation called *personalizable partial automation* that provides more comprehensive assistance for players with motor impairments using this equal partner pattern. In personalizable partial automation, players control whichever game inputs they can and control of the rest is assigned to an AI copilot.

To demonstrate this novel form of partial automation, we paired it with interface adaptation in a universally designed exergame to make the game's interaction demands more flexible for players with spinal cord injury who, as explained in Chapter 2, can have a range of motor impairments depending on the location and severity of their injury. People with spinal cord injury need to exercise to improve and maintain their quality of life, but routine exercises and exercises that patients can only perform passively may be unpleasant. Exergames may improve patients' experiences of performing these exercises, but only if the game is accessible to them.

When making rehabilitation games accessible, designers must be careful not to remove the exercise from the game. In the next section, we describe our study design and explain how we used personalizable partial automation to make two existing exergames accessible to patients in spinal cord injury rehabilitation. The primary research question addressed in this user evaluation is **RQ2**: “*Does personalizable partial automation make games accessible to players with radically different motor abilities while remaining fun to play?*”

## 6.1 Study Design

When a person’s spinal cord is injured, their sensorimotor function is impaired below the point of injury [161; 254; 140] and they need life-long rehabilitation to improve their health and independence [215]. Spinal cord injury results in either tetraplegia or paraplegia, depending on which portions of the spinal cord are injured [161]. Paraplegia is caused by injury to the sacral, lumbar, or thoracic segments of the spine and impairs motor function in the legs, pelvis, and trunk. Tetraplegia, which is caused by injury to the cervical spine, results in impaired motor function in the legs, pelvis, trunk, and arms. The extent to which motor function is impaired varies by person, ranging from minor impairment to complete paralysis in the affected regions. The American Spinal Injury Association Impairment Scale (AIS) classifies the sensory and motor impairment resulting from spinal cord injury as either complete or incomplete, assigning a letter grade from A (i.e., complete loss of sensation and motor function) to E (i.e., normal sensory and motor function) [254]. A person with a new injury is typically admitted to a hospital for inpatient rehabilitation in the months following their injury.



(a) The MOTomed viva2 passive cycling device produced by RECK.

(b) A man with tetraplegia using the MOTomed viva2.

Figure 6.1: A cycling device commonly used in spinal cord injury rehabilitation.

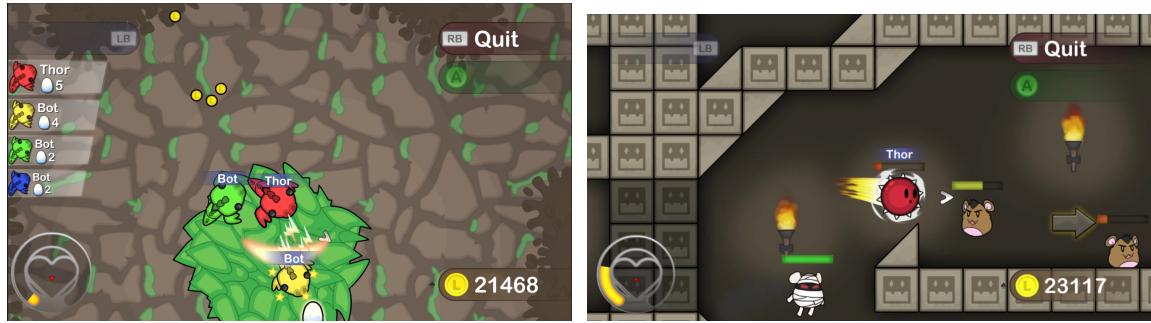
Immediately after injury, persons with spinal cord injury can improve their sensorimotor function through exercise [140; 129]. After one year, some enter neurological stability, meaning that no further improvement will be made, although persons with incomplete injuries may continue to improve [287; 123]. Exercises such as walking, cycling, rowing, wheelchair ergometry, and arm ergometry are used in inpatient rehabilitation to improve patients' functional performance while they still can. However, many persons with spinal cord injury have complete paralysis in their legs and are unable to actively contribute to lower-limb exercises. For these patients, passive range-of-motion exercises are used to provide therapeutic activation of their leg muscles [215]. These exercises are performed by either a caregiver, such as a physiatrist or physiotherapist who moves the patient's legs manually, or a motorized device, such as the *MOTomed viva2* cycling device (Figure 6.1), that moves the patient's legs mechanically. Devices such as the viva2 are widely used

in spinal cord injury rehabilitation; however, cycling may be seen as boring by patients, regardless of whether they can pedal for themselves.

We performed an exploratory study to find out whether a combination of universal design, interface adaptation, and partial automation made two exergames accessible to six participants with vastly different physical abilities due to spinal cord injury. All participants were outpatients at Providence Care Hospital in Kingston, Ontario, where the study was conducted. All participants also had prior experience using the *viva2* motorized cycling device, which moves patients' legs during rehabilitation if the patients cannot pedal. This study was approved by the research ethics boards of all institutions involved (Appendix A). As we will fully explain in Section 6.1.5, participants played two cycling-based exergames with personalized levels of partial automation. The study's key data sources were responses to a semi-structured interview conducted at the end of each session and game log files capturing participants' ability to play. Transcripts of the interviews were analyzed using reflexive thematic analysis [30; 31; 32] to identify themes between participants' reported experiences. Before discussing the participants and what they said about playing with partial automation, we will first describe how we combined universal design, interface adaptation, and partial automation in the two partially automated study games.

### 6.1.1 Partially Automated Games: *Dino Dash* and *Dozo Quest*

We implemented personalizable partial automation in *Dino Dash* and *Dozo Quest*, two games from the *Liberi* suite described in Chapter 3, now targeted to support rehabilitation of spinal cord injury. These are fast-paced action games presented in



(a) *Dino Dash*. Players gather eggs and bring them back to their nest. The player is controlling a red dinosaur that is shouting, stunning the yellow dinosaur in front of it.

(b) *Dozo Quest*. The player navigates a maze to defeat the final boss. The player is controlling the red ball to hit a “mufu” enemy with its spin dash attack.

Figure 6.2: The two partially automated exergames used in the study.

an amusing cartoon style. In *Dino Dash*, players compete to collect the most eggs and bring them to their nest. In *Dozo Quest*, players navigate a desert dungeon, defeating enemies, until they arrive at a final “boss” enemy.

These games were designed to be played using a stationary cycling device and a gamepad controller, both of which may be inaccessible to players with spinal cord injury. These devices control the game’s three inputs, which we will refer to as *Movement*, *Direction*, and *Action*. Movement determines the avatar’s movement speed and is controlled by pedalling the cycling device. Direction determines the avatar’s movement direction and is controlled with the gamepad’s left analog stick. Action makes the avatar perform a context-sensitive action in the game, such as jumping or attacking, and it is controlled by pressing any of the gamepad’s face buttons (e.g., “A” or “B”). We will now explain how these devices are used to play *Dino Dash* and *Dozo Quest*.

*Dino Dash* is an action game where players control a colourful dinosaur that collects eggs and brings them back to its nest. Red, yellow, green, and blue dinos



Figure 6.3: A man with complete paraplegia playing *Dozo Quest* using the gamepad to provide Direction and Action inputs. The viva2 is pedalling, and so copilot provides Movement input.

chase each other around the game’s arena, stunning the others with projectiles and stealing their eggs (Figure 6.2a). Players pedal the bike to make their dino move and steer using the left analog stick to avoid patches of mud or line up shots. After the player has pedalled quickly for some time, they can press any of the gamepad’s face buttons to make the dino perform a “shout” that briefly stuns opponents in front of it, causing them to drop their eggs. The first player to collect ten eggs wins.

In *Dozo Quest*, the player explores a dungeon, clashing with enemies along the way, to find and defeat a final boss. The player’s avatar is a spiky red ball, called a dozo, that can roll along the ground and do a dash attack to hurt enemies. The faster the player pedals the bike, the faster the dozo moves. Its movement direction is controlled with an analog stick. To defeat enemies that float above the ground, the player must pedal quickly, tilt the analog stick upwards to jump, and press any gamepad face button to dash (Figure 6.2b). Should the player’s dozo run out of health points, it is resurrected at an earlier checkpoint.

Due to simplicity of the *Liberi* exergames' interface, which was designed to accommodate the needs of players with motor impairments, these games may already be accessible to some players with spinal cord injury. Some players may be able to cycle quickly enough to reach their avatar's top speed and manipulate the gamepad precisely enough to hit opponents. Compared to other exergames for rehabilitation, which involve a range of body movements, playing the *Liberi* games requires only the ability to pedal a cycling device and use a gamepad. However, as explained in Chapter 3, a game's accessibility has limits and even games designed to be broadly accessible may exclude some players. In the following sections, we explain how we combined interface adaptation and partial automation in the universally designed *Liberi* exergames to extend access to players with a broader range of motor abilities.

### 6.1.2 Universal Design in *Dino Dash & Dozo Quest*

The *Liberi* exergames were designed for children with cerebral palsy and are therefore able to accommodate players with significant motor impairment. In accordance with universal design principles, controlling the games involves abilities that are widely held by target players: the ability to pedal a specially designed bicycle, the ability to use an analog stick, and the ability to press a button. These games were chosen for our study because this range of motor abilities is held by some but not all persons with spinal cord injury. Some players would be unable to play without adapting the games, allowing us to determine whether partial automation made them accessible. Design choices, such as having each of the gamepad's face buttons trigger actions, may make *Liberi* more accessible than other exergames for players

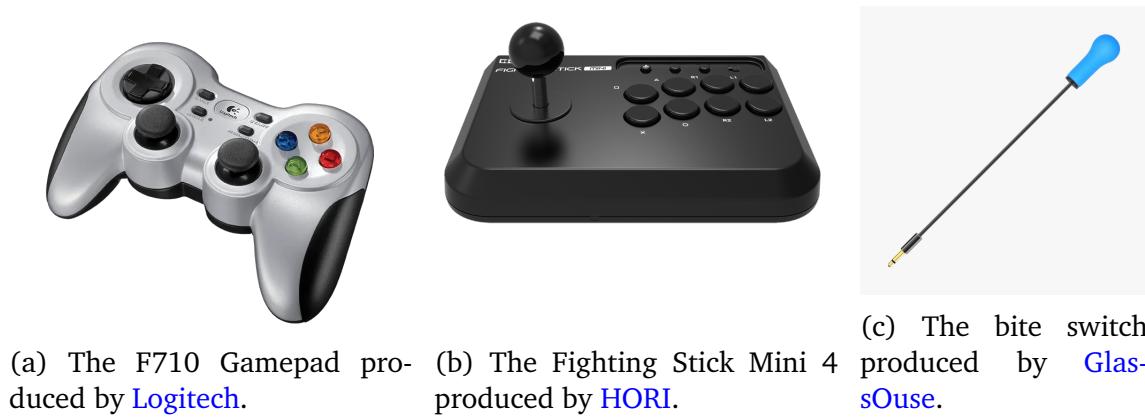


Figure 6.4: The devices used by participants.

with spinal cord injury, but we knew that the involvement of pedalling would make it inaccessible to many patients with spinal cord injury.

### 6.1.3 Interface Adaptation in *Dino Dash & Dozo Quest*

To overcome the accessibility limitations of *Dino Dash* and *Dozo Quest* for players with spinal cord injury, we used interface adaptation to allow players greater flexibility in how they play, a strategy suggested by our ontology of games accessibility approaches in Chapter 3. Specifically, players who could not use the gamepad (Figure 6.4a) were offered a joystick (Figure 6.4b) to replace the analog stick and a bite switch—a button held in and activated by the player’s mouth (Figure 6.4c)—to replace the gamepad buttons. To support use in spinal cord injury rehabilitation, the games were set up to be played with the widely used MOTOMed viva2 cycling device, which affords active cycling to patients who can pedal and passive cycling to patients who cannot. Players who could not actively cycle instead played while



Figure 6.5: A woman with incomplete tetraplegia playing *Dino Dash* using a joystick and bite switch to provide Direction and Action inputs. The viva2 is pedalling for her, and so the copilot provides Movement input.

passively cycling. As mentioned in Chapter 3, designers using partial automation should take care not to remove the exercise component of rehabilitation games. Since pedalling the viva2 may be performed either by the user or by its built-in motor, automating the Movement input in *Dino Dash* and *Dozo Quest* does not diminish the utility of these games in spinal cord injury rehabilitation.

Some people with spinal cord injury can pedal a bicycle and use a gamepad controller, and so can provide their own Movement, Direction and Action inputs using these devices. Others have paralysis below the neck and can perform their own Action inputs with a bite switch, but they may require assistance in controlling the game's other inputs. In this way, players use different devices to play the games, depending on their physical abilities. Our strategy is that if an input can be made accessible by substituting a different input device, we do so; for example, substituting a bite switch for a button on a controller. However, we expected that not all players with spinal cord injury would be able to use these alternative devices and may require further support to play. If no device made an input accessible to a



Figure 6.6: A man using a MOTOrmed viva2 (left of image) for rehabilitation of spinal cord injury. The viva2 is a pedalling device where the users' feet are secured to the device while they sit in their own wheelchairs. The device's motor moves the pedals for users who are unable to pedal for themselves.

specific player, then we assigned control of that input to personalizable partial automation. Partial automation is therefore a last resort, to be used when traditional accessibility techniques are insufficient.

#### 6.1.4 Personalizable Partial Automation in *Dino Dash & Dozo Quest*

Building on universal design and interface adaptation, the final step in making *Dino Dash* and *Dozo Quest* more accessible is partial automation, as suggested by our ontology of games accessibility approaches in Chapter 3. The novel form of partial automation used in these games is called *personalizable* partial automation, because it is designed to further personalize players' control of games. In personalizable partial automation, if a player is unable to control a game input, then control of that input is assigned to an AI copilot that shares control with the player (Figure 6.7). In accordance with the equal partner shared control design pattern described in

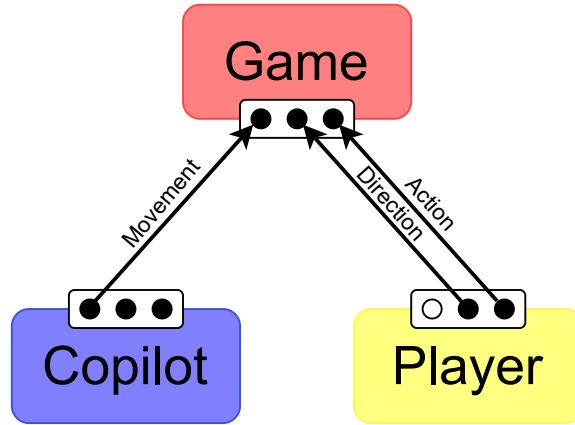


Figure 6.7: Partial automation: The game interface offers inputs for controlling movement, direction, and game actions. The player controls as many of these inputs as they can (in this example, Action and Direction). An AI copilot provides inputs for parts of the interface that the player cannot control (in this example, Movement).

Chapter 5, the copilot plays a *Complementary* role and its control signal is *Combined* with the player's in an *Unsupervised* and *Independent* manner. This enables players to select the inputs they control while playing and relieves them of tasks related to the inputs that they do not control.

To achieve this, we created custom AI copilots to automate play of each game. The copilots operate by injecting inputs into the game mimicking inputs that a player would provide. Both copilots were implemented at the level of the games' source code and have complete knowledge of the games' states. To be successful, these copilots need to follow strategies that play the game well and that perform actions that the player would expect.

The *Dino Dash* copilot uses classic game AI steering behaviour [205] to navigate the play area using the Direction input. The copilot transitions between behaviour states depending on its surroundings. For example, when a nearby dino is carrying an egg, and there are no free eggs nearby, the copilot will transition to *chase mode*

and prioritize chasing down the dino to steal its egg. Attractive and repulsive forces exerted by game objects determine how fast the avatar moves using the Movement input. For example, a free egg exerts a large attractive force that causes the avatar to move at full speed towards it. If the avatar is not carrying an egg and encounters another dino within a short range ahead, the copilot uses its shout to stun the opponent by triggering the Action input. These behaviours enable the copilot to independently control any subset of the game’s three inputs, no matter which inputs are controlled by the player.

The *Dozo Quest* copilot moves from room to room in search of enemies, targeting whichever enemy is closest. When a new enemy is targeted or the dozo is knocked off course, the copilot uses the A\* pathfinding algorithm [205] to plan a path from its current location to its target’s location. The copilot uses the Direction input to direct its movement along the path and the Movement input to roll along at top speed. Once in range of one or more enemies, the copilot aims directly at the closest one, gets closer, and dashes at it using the Action input. Sharing control of *Dozo Quest* involves both the player and the copilot coordinating their control of the Movement, Direction, and Action inputs to perform complex actions like the dash attack. No matter which inputs are controlled by the player or the copilot, both need to coordinate their actions to play effectively.

### 6.1.5 Combining Approaches in *Dino Dash & Dozo Quest*

These examples of *Dino Dash* and *Dozo Quest* show that it is possible to develop games that use partial automation with the goal of increasing the games’ accessibility. With these games, we have demonstrated that it is possible to start with a

base of universal design, extend accessibility using interface adaptation, and then, finally, use partial automation only when these techniques are insufficient. This allows a game to have full features available to persons who can use a game controller and a pedalling device, while still allowing play by people who cannot use a joystick, or who rely on the viva2's passive pedalling.

We hypothesized that partial automation would make *Dino Dash* and *Dozo Quest* accessible to players with vastly different abilities due to spinal cord injury. For example, Figure 6.5 shows a player using a personalized configuration of input devices and automation. She uses the joystick to control the Direction input, and uses the bite switch to provide Action inputs. A second player, shown in Figure 6.3, has complete paraplegia and cannot pedal the viva2 to control the Movement input. This player uses the gamepad to control Direction and Action, with the Movement input automated. A third player (Figure 6.6) has complete tetraplegia, and plays using a bite switch to control the Action input, while the Movement and Direction inputs are controlled by the copilot. We will now describe our participants and the alternative interfaces they used to play the study games.

### 6.1.6 Participants

Six participants were recruited and successfully completed the study procedure. Participants were required to have spinal cord injury, be a patient at the hospital, be 18 to 50 years of age, have at least fifty hours of lifetime gaming experience, and be able to engage in an interview. For this initial evaluation, we excluded non-gamers to focus on the perspectives of players who are disabled by games solely due to motor impairment. They were recruited by a spinal cord injury physiatrist at a local

Table 6.1: Participants' demographic information. AIS grades indicate whether their injury is complete, meaning no sensory or motor function was preserved at sacral segments S4-S5, or incomplete, meaning some sensory or motor function was retained. NLI (Neurological Level of Injury) specifies the part of the spine that was injured.

Code	Age	Sex	Years Injured	AIS Grade	NLI
P1	31	M	< 1	Incomplete Paraplegia (C)	T11
P2	33	M	5	Incomplete Tetraplegia (B)	C4
P3	31	M	< 1	Complete Paraplegia (A)	T5
P4	23	M	2	Complete Paraplegia (A)	T4
P5	33	F	15	Incomplete Tetraplegia (B)	C5
P6	28	M	9	Complete Tetraplegia (A)	C4

hospital and via a poster that was circulated to members of the Spinal Cord Injury Ontario Peer Connections community organization. Each participant met with us individually in the hospital's outpatient gym for a single 90 minute session.

Table 6.1 shows the participants' demographic information and Table 6.2 shows participants' gaming information including the devices they used and the game inputs they controlled during play. All participants were out-patients who had completed their in-patient rehabilitation. In spinal cord injury rehabilitation, patients' motor and sensory abilities are classified using the ASIA impairment scale (AIS) [254]. A patient's AIS letter grade indicates their level of motor function below where they were injured (neurological level of injury – NLI), ranging from A to E. An E indicates normal motor function, D and C mean some impairment, and patients with B and A have complete impairment. Participants' information indicates that they had vastly different physical abilities, ranging from AIS grade C paraplegia where the participant could pedal a bicycle and use a standard game controller,

Table 6.2: Participants’ gaming information.  $GF_<$  and  $GF_>$  denote participants’ gaming frequency before and after injury respectively. The Cycling column indicates whether participants actively or passively cycled during play. The Movement, Direction, and Action columns indicate the devices participants used to control each input (“V”=viva2, “G”=gamepad, “J”=joystick, “B”=bite switch; no device indicator means input was automated).

Code	$GF_<$	$GF_>$	Cycling	Movement	Direction	Action
P1	Monthly	Monthly	Active	V	G	G
P2	Daily	Weekly	Active	V	G	G
P3	Weekly	Monthly	Passive		G	G
P4	Daily	Daily	Passive		G	G
P5	Daily	Daily	Passive		J	B
P6	Weekly	Monthly	Passive			B

through to AIS grade A tetraplegia where the participant could interact with the games using only a bite switch.

### 6.1.7 Method

Participants were invited to the hospital’s outpatient gym to play three games: *Dino Dash*, *Dozo Quest*, and a cycling-based rehabilitation game provided with the viva2 called *MOTOMax* (Figure 6.8). The *Liberi* exergames used in the study were included to test the efficacy of personalizable partial automation while *MOTOMax* was used in this study to provide a baseline for current commercial spinal cord injury rehabilitation games and to elicit participants’ impressions of a rehabilitation game that is inaccessible to some persons with spinal cord injury.

In *MOTOMax*, players control a circular avatar that moves to the left or right side of the screen when the player is pedaling harder with their left or right leg

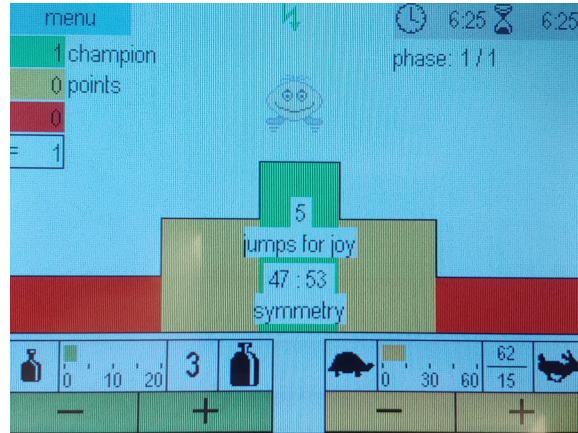


Figure 6.8: The *MOTOMax* game shipped with the viva2 rewards players for symmetric pedalling.

respectively. These relative forces are shown on-screen to the player as the percentage of the pedalling force exerted by each leg. When the player exerts symmetric pedaling force, the avatar jumps up and down in the center of the screen, awarding the player points based on how fast they are pedalling. Although this game can be played while the viva2 passively pedals, the player’s score will never increase. This is because the point of the game is to promote effortful and symmetric pedalling while passive cycling is inherently symmetric and effortless.

Upon arrival, participants were guided through the study’s informed consent procedure and asked to complete a demographic questionnaire. Participants who were unable to sign the consent form or fill out questionnaires were assisted by their care worker or the researcher conducting the session. Participants’ physical abilities were assessed by the physiatrist. They were first asked if they could use the *Liberi* games’ default devices. If not, participants were provided with a personalized interface, and inputs that participants could not control with any device were automated. Each participant was asked whether they could pedal the viva2

and use a Logitech F710 gamepad (Figure 6.4a). If they could not pedal the viva2, the device's motor pedalled for them and the Movement input was automated. If participants did not believe they could use the gamepad, they were asked if they could use a HORI Fighting Stick Mini 4 joystick (Figure 6.4b) to provide Direction input and a GlassOuse bite switch to provide Action input (Figures 6.4c). Inputs that participants could not control with these alternative devices were automated. For participants who were unable to actively cycle, *MOTOMax* was played while passively cycling. Participants played the *Liberi* exergames with a selection of accessible devices matched to their abilities, listed in Table 6.2. Before playing each exergame, the participant's physical condition was assessed by the physiatrist to confirm their fitness to continue.

Participants then played each exergame for approximately five minutes. We explained to participants the games' mechanics and goals, as well as how the inputs under their control affect their avatar's activities. They first played a warm-up round of each game to verify that they understood how to play. We did not tell them that control over some of the avatar's activities would be automated or explain how the games were played using the inputs that they did not control. This was done to determine whether participants could understand how to play the games using only the inputs that were accessible to them. Participants played one round of *MOTOMax* followed by two rounds of *Dino Dash* and then one round of *Dozo Quest*, totalling roughly five minutes for each game. Following play of all of the rehabilitation games, participants engaged in a semi-structured interview about their experiences. Although participants played each game for only five minutes, this was sufficient to

determine whether partial automation made the games accessible and yielded rich interview data for qualitative analysis.

### 6.1.8 Data Collection

Three forms of data were collected during sessions: (1) a demographic questionnaire, (2) gameplay logs and video of gameplay, and (3) a semi-structured interview. The demographic questionnaire recorded participants' age, sex, AIS classification, neurological level of injury (NLI), and their gaming experience both before and after their injury. During play of the *Liberi* exergames, successful performance of key game activities was recorded. In *Dino Dash*, these were *moving*, *steering*, *picking up an egg*, *scoring a point*, and *hitting another Dino with a shout*. In *Dozo Quest*, recorded events were *moving*, *steering*, and *hitting an enemy with a spin dash*. The presence of these events in the gameplay logs shows the degree to which participants were capable of engaging in all aspects of play.

Following play of both exergames, participants were interviewed for approximately thirty minutes about their overall experience and experience playing each of the exergames. We asked them whether they liked each game and whether they experienced fun, difficulty, or frustration while playing. We also asked about significant moments during play, whether they liked the pace of the games, and whether they felt like they were in control while playing. These interviews were video-recorded to allow later analysis.

Table 6.3: Players abilities to carry out gameplay actions in the rehabilitation games. A ✓ indicates that the participant performed the action at least once during gameplay. A ~ indicates that an error prevented gameplay data collection.

	P1	P2	P3	P4	P5	P6
<i>Dino Dash</i>						
Moved	✓	✓	✓	✓	✓	✓
Steered	✓	✓	✓	✓	✓	✓
Hit an opponent	✓	✓	✓	✓	✓	✓
Picked up an egg	✓	✓	✓	✓	✓	✓
Scored a point	✓	✓	✓	✓	✓	✓
<i>Dozo Quest</i>						
Moved	✓	✓	~	✓	✓	✓
Steered	✓	✓	~	✓	✓	✓
Hit an enemy	✓	✓	~	✓	✓	✓

## 6.2 Results: Ability to Play

As shown in Table 6.2, there are multiple actions players need to be able to perform to play *Dino Dash* and *Dozo Quest*. So, for the purpose of analyzing participants' gameplay data, we define being able to play as being able to perform these actions. We found that the use of interface adaptation and partial automation allowed participants who would have been disabled by the games' default interface to play in a different way. P5 and P6 were unable to use a game controller, but benefited from alternative devices; P5 was able to control the Direction input with the joystick, and both were able to control the Action input with the bite switch. Partial automation extended the improved accessibility provided by interface adaptation even further. Four of six participants were unable to pedal the viva2 but were able to play using passive pedalling and the assistance of partial automation to determine when

and how fast their avatar should move. The accessibility of these universally designed games, now enhanced with interface adaptation and partial automation, shows how these three separate techniques can be combined to make a game accessible to players with motor impairments ranging from minor deficits in manual dexterity to complete paralysis below the neck. In contrast, only P1 and P2 were able to play *MOTOMax*, which affords no personalization.

We manually reviewed recordings of participants' gameplay to determine which actions they were capable of performing, either on their own or with the assistance of partial automation. These data, captured in Table 6.3, show that with the assistance of partial automation all participants were able to use all features of *Dino Dash* and *Dozo Quest*. This included core functions such as moving the avatar and game mechanics such as picking up objects and attacking enemies. From this, we conclude that the use of partial automation to personalize *Dino Dash* and *Dozo Quest* was successful in extending access to full-featured games. We found that participants with vastly different physical abilities, ranging from people who could pedal a bicycle and use a gamepad to people who were capable of interaction only through a single bite switch, were able to play both games. This establishes that for these two games, partial automation met its primary objective of making the games accessible to people with vastly different physical abilities. This success in increasing accessibility, however, does not reveal whether players enjoyed partial automation, or were even aware of its effects. The next section reports on players' experiences with partial automation.

### 6.3 Results: Player Experience

One member of the research team watched and transcribed video of participants' sessions and semi-structured interviews. Transcripts were analyzed using reflexive thematic analysis from a realist perspective informed by the gameplay logs. *Critical realism* is one of the epistemological orientations researchers can assume in doing thematic analyses [32]; this means that we believe participants discussed an external reality (i.e., what happened in the games) that we can know about. Therefore, participants' statements about gameplay events, like winning the games or strange copilot behaviour, were compared with gameplay recordings. This was done to discover how partial automation influenced participants' experiences. Transcripts were coded inductively with consideration for both their explicit semantic content (i.e., *what they said*) and their implicit latent content (i.e., *why they said it*). In particular, we were interested in participants' reported experiences of using partial automation. One researcher coded the data and proposed an initial set of themes which were then discussed in depth with two other researchers while making reference to the original transcripts. Through this process, we identified four major themes, which over several meetings were refined and reorganized into major themes with minor sub-themes.

Participants' experiences of playing exergames with partial automation were strongly positive. They said that partial automation enabled them to play exergames that would have been inaccessible otherwise and also that playing these games gave them a new way to participate in their rehabilitation. Participants found that the games were far more accessible to them than other games they had tried to play

since their injuries. Due to their prior experiences of being disabled by games, participants were sensitive to the accessibility needs of others with different abilities. They believed that partial automation would greatly increase the accessibility of exergames for others with motor impairments. As stated by P5: “[Partial automation] *opens up an avenue for people with different types of disabilities and different types of abilities to be able to play.*”

Although we thought that addressing the boredom of rehabilitation exercise, described in Chapter 2, was the primary way in which games could improve patients’ experiences of rehabilitation it became apparent that the fun and motivation that rehabilitation gaming affords was only a small part of the stories participants had to tell. Their responses to questions such as “*Did playing the game make you focus on using the [viva2]*” were far more detailed and introspective than we anticipated. While participants did find play more compelling than their normal exercise, evidently this was not the only way that play was meaningful to them. Their accounts of how rehabilitation made them feel and how playing games, in physiotherapy or for leisure, might have made these experiences better indicate that rehabilitation games may provide patients benefits beyond making exercise more fun. As stated by P6: “[Rehabilitation gaming would be beneficial] *for anyone who, I’d say, has had some sort of injury and thinks that life might be over, but there’s still lots to do in life.*” In this section, we will first present one major theme related to participants’ beliefs about the benefits of accessible rehabilitation gaming and then present two major themes related to participants’ experiences of playing with partial automation.

### 6.3.1 The Benefits of Accessible Rehabilitation Gaming

A traumatic spinal cord injury can be a life-altering event that prevents the person who is injured from living the life they lived before. Bourke et al. analyzed interviews with patients with tetraplegia, who were recently discharged from inpatient rehabilitation, and found that they saw spinal cord injury as a “*biographical disruption*.” [29] Other researchers, as well, have reported that patients may believe that the lives they knew before their injury are over and that they may feel out of place in their changed bodies [317]. Patients may come to distinguish their “*nondisabled*”, “*internal*” selves from their newly “*disabled*”, “*external*” selves [317]. This can make rehabilitation a depressing experience, filled with ruminations about activities that are no longer accessible and uncertainty about the future, that leaves patients feeling alienated by others and from themselves [317; 50]. For patients to reintegrate themselves into their bodies, they need to be motivated to overcome the challenges of life outside rehabilitation, to reconnect their past lives with the present, and to envision a future that is better than the bleak situation they find themselves in [317; 186]. Caring rehabilitation staff, supportive peers with spinal cord injury, and leisure activities can help patients make this transition by enabling them to discover what living with spinal cord injury means and showing them that they can still enjoy their lives [50].

Our participants shared similar perspectives. The major theme *from passive to active* illustrates how accessible gaming might have improved participants’ rehabilitation experiences.

### 6.3.1.1 Major Theme: *From Passive to Active*

Playing games while cycling gave participants new ways to engage in rehabilitation exercises, made their efforts more meaningful and rewarding, and enabled them to actively participate in a previously disabling activity. However, participants' experiences of each game differed greatly depending on whether the game was accessible. *MOTOMax*, which was inaccessible to P3-6 because it required them to actively cycle, conferred benefits to P1 and P2 but not others. In contrast, all participants said that playing the universally accessible *Dino Dash* and *Dozo Quest* games would have improved their experiences of rehabilitation. The following minor themes recount how playing rehabilitation games improved participants' experiences of cycling rehabilitation therapy. The minor themes are: *gaming engages the whole body*, *gaming engages body and mind*, *gaming gives meaning to exercise*, *gaming lets you win at something*, and *gaming enables active participation in rehabilitation*.

**Minor Theme: *Gaming engages the whole body*** We began each interview by asking participants to compare playing rehabilitation games with their normal physiotherapy. For participants who could not pedal, the games gave them new ways to engage in cycling. P5 explained that playing *Dino Dash* and *Dozo Quest* put her mind at ease during an otherwise uncomfortably boring activity: “*I haven’t got to work out my legs in a while. So, it was nice but it also gave me something to... I’m one of those people that I have what I have labelled ‘idle hand syndrome’.*” She went on to say that “*It’s more so in a sense that I can be doing something but if my brain isn’t engaged, if I’m not busy doing something else... Like, I like to multitask a lot.*” She, and other participants, wanted something to do with their idle hands and minds while cycling.

P4 was particularly excited about the possibility of playing virtual reality (VR) games in physiotherapy. He explained how he plays a game called *Subnautica* that asks players to perform large arm movements to control the game's mechanics. He explained that "*If you play it in VR, you have to move your arms to do a swim motion... You can't get your legs moving if you're a paraplegic, like myself, but it gets your core muscles and upper-body going.*" He believed that VR games could improve rehabilitation for patients with paraplegia who could engage in gamified upper-body exercises, but not leg exercises.

Conversely, P1 was able to pedal the viva2 and use the gamepad. When asked how *Dino Dash* could be made more enjoyable, he said that he might have preferred to play without the controller. He was especially invested in doing high quality exercise and believed that focusing on cycling would improve his workout. Therefore, he had reservations about using a gamepad while cycling, since it might diminish his quality of exercise. However, eventually he concluded, "*I guess the joysticks are better. Gets everything working: brain, hands, legs.*" Playing games while cycling offered participants a holistic means of participating in their exercise.

**Minor Theme: Gaming engages body and mind** Playing rehabilitation games gave participants something to focus on while cycling, which engaged not only their bodies but their minds as well. P5 said, "*It engages your brain as you're engaging your body as well and it doesn't really make it seem like working out. It makes it fun.*" She explained that gaming provided an interactivity that other forms of entertainment could not, saying that "*It was fun to actually focus on something other than like just listening to music, like my brain could actually*

*interacting.*” P2 shared a similar sentiment, believing that distracting patients from boring exercises would be beneficial: “*If I had the ability to listen to music in that moment then I would. But, then you’re still focusing on the exercise itself. It’s a way, I think, having a game in front of you or a visual aid of some kind, especially if it’s interactive with the exercise itself, would take your mind off the exercise itself completely.*” P6 saw the potential for rehabilitation games to engage patients in their exercise. He said that “*It definitely is a way for me to keep myself, like, engaged in actually what is going on and I think it is, for myself as someone who hasn’t done a lot of gaming, it is an interesting way to engage people into their therapy sessions.*”

The games made doing the same repetitive and boring exercise more engaging by giving participants new virtual worlds and ways of playing to explore. P4 liked that playing *Dino Dash* required him to keep track of and reason about many game pieces. He said that “[Playing *Dino Dash*] made my brain think like ‘Okay, I’ve gotta focus on three other characters, I gotta focus on where the eggs are gonna spawn, where I gotta deliver them to, as well as my power level, and when I can attack.’” P2 believed that this increased cognitive load could reframe rehabilitation exercises as a primarily mentally stimulating activity. He explained that “*You’re not even thinking about the exercise you’re getting and now you’re training your brain as well. So, you can literally play that game for half an hour and not realize you just biked for half an hour. So, it doesn’t even feel like exercise... Sure, there’s exercise involved in it, but I’m not exercising, I’m playing a game. So, it’s different. It’s a different mindset.*”

Participants found playing games to be a welcome distraction from cycling, which could be mentally under-stimulating and induce rumination. Playing the games engendered a different mindset that helped participants to overcome the discomforting boredom, anxiety, and disability that they normally experience while cycling. For participants who cycle passively, play made them feel better about being unable to contribute to their exercise. However, games can only benefit patients when they are able to play. *MOTOMax* was inaccessible to P4, since he passively cycled during play, and therefore it did not hold his attention. He said that “[*MOTOMax*] is the one that I liked the least, ‘cause I had no way of actually doing anything... Because the game relied on you pedalling equally on both sides to complete it. I can’t pedal.” Disabled by the game, P4 said that being unable to play *MOTOMax* was frustrating and that the cycling was no different than normal. He spent this part of his session waiting for it to be over and thinking “*Can we just move on to the next, please?*”

**Minor Theme: *Gaming gives meaning to exercise*** Participants found that playing games was gratifying in ways that rehabilitation exercise was not. P6 believed that rehabilitation gaming could help patients to find the motivation to exercise. He said, “*some people might feel that it is a waste of time and not sure why they’re doing it... So, I think it just allows them to have another objective to why they are doing physio.*” Participants who could not pedal believed that patients would be motivated to exercise if they made overcoming the games’ challenges a rehabilitation goal. P5 explained that “[Gaming would be motivating] just because it would give you something to look forward to... Then if there is more levels, then obviously people will want to keep coming back to play

*more levels to see how far that they can get.*" Although patients' achievements in the games may be unrelated to the exercise they are doing, participants who could not pedal believed that gameplay related rehabilitation goals would compel patients to exercise and would make their efforts more meaningful.

P1 and P2 believed that patients who could pedal would also be motivated by exercise-related gameplay goals. P2 proposed that a fake scoreboard be added to *MOTOMax*. He told us to "*Make the top score slightly under [the player's best] so that whenever anybody's doing it they can feel that more of a sense of accomplishment. Because otherwise the number you get at the end is meaningless without merit. Like, so great, I got an 82. Is that normal? Is that above average? Below average? If you gave it meaning to the player I think that would make it more enjoyable to them. I have to beat this number.*" He believed that feedback from *MOTOMax* would enable competitions between patients in rehabilitation, motivating them to work harder. P1 found an immediate benefit to *MOTOMax*'s feedback. He said that "*It'd be a goal... Like while you're on it, just to see and check both legs... What leg's stronger, what leg's weaker. You're pushing to keep it in the middle.*" During play, the game informed him of how hard he was pedalling with each leg. This provided him with immediate feedback, which P2 said was sorely lacking in rehabilitation. He said that "*It's just more feedback immediately rather than waiting for a doctor to say this or that. You'd automatically see how you're doing.*" He believed that feedback from *MOTOMax* would be a valuable tool for patients to track their progress over time, motivating them to "*every day do better.*"

Participants believed that rehabilitation gaming would motivate patients to exercise, but only if the games were accessible. When asked if she believed that playing *MOTOMax* would make physiotherapy more enjoyable, P5, who passively cycled during play, said “*I don’t think I could really give an opinion on it because it didn’t work for me. So, I don’t know how the game is supposed to work, but if someone has like a little bit of, you know, ability to move their legs it might definitely help them in encouraging them to do more possibly. Because they can see how much they’re like contributing to the game.*” *MOTOMax*’s exercise performance feedback benefits players who can pedal at the expense of players who cannot. The feedback that P1 and P2 found so useful served only to reconfirm for P3 that he was disabled by active cycling. When we discussed *MOTOMax* he said, “*I was just watching to see if I was actually pedalling at all. So, I was just concentrating on if I was doing anything, which I wasn’t.*”

**Minor Theme: Gaming lets you win at something** When asked if any significant moments occurred while playing *Dino Dash* and *Dozo Quest*, participants often cited their victories. Competing against AI-controlled enemies and winning the games was thrilling in ways that cycling was not and gave participants “*a little bit of a sense of accomplishment*”, as P5 put it. P4 said that “*It was sort of putting on tension and a rush like ‘Okay, now I gotta beat this guy. I can’t lose.’*” The games’ challenges gave players obstacles to overcome and made them better players as they gained more experience. Becoming more competent players and beating more challenging opponents validated participants’ efforts. P4 recounted how this made him feel while playing *Dino Dash*, “*I was actively seeing progression, myself getting better and better. Winning like that*

*felt good.*" For participants who could not actively pedal the viva2, gaming offered them a chance at victory, which cycling could not.

The opportunity to win is what P3 liked best about *Dino Dash*. He said, "*It was a lot of fun... Just 'cause it was like a race... Like, how you can win at something... Puts a little competition into it.*" P2, as well, found the game's competitive nature compelling and believed that competitions between patients would motivate them to exercise. He said "*Something like this would be great just to socialize you with your other patients, other people, and just get you going again, and give you a competitive fire again, and maybe make you want to work harder.*" He believed that social interactions with competitors would enable patients to help each other cope with feelings of hopelessness. He explained that "*[Patients become] upset when somebody is just giving up. Like I yelled at a bunch of people in physio because they would just quit and something like this where you could literally stack up against everybody [would be motivating.]*" P2 also saw opportunities for clinicians to leverage patients' envy of each others' abilities to drive competition. He said, "*The one thing that a lot of the doctors and physios don't take into account is the human aspect. The fact that each patient is looking at every other patient as well and wishing 'Oh, they can do this; so, I wish I could do this. They can do that; I wish I could do that.'*" He believed that playing *Dino Dash* against other patients would have motivated him to spend more time exercising as an inpatient. He said that "*I would have been on that thing every free moment I had, after all my therapies and whatever, I would have been going to Dino Dash. Gonna kick some ass!*"

**Minor Theme: Gaming enables active participation in rehabilitation** Participants

who could not pedal the viva2 believed that they were not participating in cycling, since the device's motor moved their legs for them. P3 described how this made him feel and explained how playing the games made him feel better. He said, "*If it's just passively pedalling then I feel like I'm not really doing anything. So, if you have something else there on the same, you know, device, it makes you feel like a little better about using it.*" For these participants, cycling represented an activity that they needed to do, despite their inability to actively participate in the exercise. When asked what he liked about *Dino Dash*, P6 said that it gave him something to do while cycling. He said, "[It was engaging] just being able to... play a game while you're there, instead of just sitting there and staring at a wall." The games not only alleviated participants' boredom while cycling, it enabled them to participate in a previously disabling activity. When describing what he liked about playing games while cycling, P3 said that "*There's a little more to do than just go through the motions of physio.*"

Participants only reported feelings of increased participation when playing games that were accessible to them. P1, who pedalled the viva2 while playing *MOTOMax*, said that it made cycling more engaging because "*You have a goal of keeping that guy in the middle. Keeping both legs at 50% and just not staring off into the world, doing nothing.*" However, P5 was unable to play *MOTOMax* and believed that, just like her normal cycling-based exercise, she "*Totally wasn't contributing to [MOTOMax] at all.*" For participants who could not play *MOTOMax*, the game offered no benefit. As P4 put it, "*To me [playing*

*MOTOMax] was just going back to regular exercise. It wasn't really pulling me in."*

Playing rehabilitation games with partial automation provided new ways for players to participate in physiotherapy. Participants believed that gaming would have motivated them to engage in otherwise demotivating and disabling exercises. Playing the games gave participants something to do while passively cycling and served as a much needed distraction from the other things that sometimes occupied patients' minds. Their reception of the partially automated games was extremely positive, as participants believed that gaming as a rehabilitation activity would have alleviated many of the discomforts they experienced as inpatients. However, only the participants who actively cycled had a favorable view of the largely inaccessible *MOTOMax*. For participants who were unable to pedal, watching the *MOTOMax* avatar stand still and receiving a final score of 0 only added insult to injury. We, therefore, conclude that the success of *Dino Dash* and *Dozo Quest* was due to the improved accessibility afforded by partial automation. Most participants would have been unable to play without it, as they explained in their interviews. In the next section, we present two more themes related to participants experiences of using partial automation.

### 6.3.2 Participants' Experiences of Partial Automation

Participants enjoyed playing rehabilitation games with partial automation and were especially pleased with the games' customizability. Pairing partial automation with interface adaptation allowed participants greater personalization of the physical interfaces they used to play. Since participants were not required to control all of the

games' inputs, they could play using whichever set of devices was accessible and familiar to them. Participants believed that partial automation would enable them to play using other computer interfaces that were not designed for gaming and that it would make games accessible to other players with different motor impairments. Overall, participants were excited about gaming as a rehabilitation activity and credited partial automation with the *Liberi* games' accessibility. Partial automation allowed participants to personalize the games' interaction demands to their abilities, but it also caused them unforeseen difficulties around cooperating with the copilot. In this section, we present two themes related to participants' experiences of partial automation: *adaptation both includes and excludes* and *automation confusion*.

#### 6.3.2.1 Major Theme: *Adaptation Both Includes And Excludes*

Participants believed that partial automation would enable players with different abilities to play the same games, but recognized that players' experiences may be different. The games' hardware interfaces and mechanics allowed participants to take advantage of their unique abilities to make play uniquely meaningful to them. However, some parts of these games indicated to participants that play may be better for others with abilities they lacked. In this section we describe two minor themes related to participants' experiences of inclusion and exclusion.

**Minor Theme: *Adaptation Makes Players Feel Able*** Playing *Dino Dash* and *Dozo Quest* with an adapted hardware interface enabled participants to play in their own ways. They liked using devices that were similar to devices they use in daily life and wanted to personalize their interfaces further using their own

devices. One unintended benefit of partial automation is that it allows players to delegate control of unwanted inputs and focus on inputs that are most important to their rehabilitation (i.e., controlling Movement by pedalling the viva2). This theme illustrates how revisiting familiar game mechanics and using familiar devices made participants feel accomplished when they were able to employ their existing skills to play.

Participants wanted to play using devices that they use in daily life and found that using familiar devices can make players feel competent. P5 has incomplete tetraplegia and drives her power wheelchair using a joystick. She showed us the trick she uses to grip long objects and said that the same trick would enable her to use the joystick to play. We placed it on the table to her right (Figure 6.5) and she said: “*I should have no problem with this.*” P6 also uses a power wheelchair and brought his own devices to the session. While playing both games with the bite switch, he tilted his head from side to side, as if to direct his avatar in that direction. During his interview, he explained how he would have liked to use his power wheelchair’s head controls to play: “*I think that would be an opportunity to use an already existing control, that the person already knows, to be able to control what they want to do on the games.*” A game that supports power wheelchair controls may have enabled P6 to leverage his unique abilities.

Beyond this desire to use familiar devices, participants found that feeling rewarded for their expertise made play meaningful to them. P4 identified *Dozo Quest* as a *Metroidvania*-style game, which made it immediately familiar to him. He believed that his expertise in this style of game enabled him to quickly

learn to play. “*I was given the tools right off the bat, and since the controls were simplistic enough I was able to pick that up quick.*” (P4) Although he described *Dozo Quest* as giving him the tools, it may be more appropriate to say that P4 brought his own tools, which *Dozo Quest* enabled him to use. P2 liked that the dozo jumped higher when he pedalled faster. He said that this provided him with feedback that indicated how well he was pedalling and rewarded him for doing more vigorous exercise. “*Getting him to jump a certain way or using the button for that burst only worked if you had that speed built up. So, the more speed you had the more lift you can get...*” (P2) The mechanical similarities that *Dozo Quest* shared with games P4 had played before and its jump mechanic, which rewarded P2 for pedalling quickly, gave them the impression of being rewarded for doing things they are good at doing.

Personalizing control of the games afforded participants greater freedom to choose how they play and automation may have enabled participants to focus on the most important aspects of play. When asked how *Dino Dash* could be improved, P1 said that he might have preferred to play without the gamepad, so he could focus more on pedalling. This answer was surprising; other participants wanted more control over the games, not less. He explained that playing *Dino Dash* distracted him from cycling. “*You’d wanna be more focused on your workout, wouldn’t it? ... Just ’cause the game, it’s distracting you...*” (P1) His priority was getting a good workout and he wanted to personalize his control of *Dino Dash* to provide the highest quality exercise.

**Minor Theme: *Adaptation Makes Players Feel Disabled*** In contrast to the positive experiences described above, participants described past negative experiences with inaccessible exergames, which raised concerns about whether the games they were testing would be accessible. They wanted exergames to be accessible both to them and also to others with different abilities. They felt disabled by previous games they could not play, but also felt disabled by games that required them to play differently from other people. One participant explained that playing with partial automation gave him the impression that he had a diminished experience of play. This theme illustrates how adaptations for players with motor impairments can make them feel disabled, even when the adaptations allowed them to play the game.

Specifically, past negative experiences with inaccessible games coloured players' expectations entering this study. P5 explained that since her injury, exergames that use the Kinect have been inaccessible to her. "*It took me like an hour using the regular remote, and like I said, to make up this avatar. And then it was like ‘please stand in front of the sensor.’ So, it was like it wouldn’t recognize the lower half of my body.*" (P5) Even though she could use the gamepad to customize her avatar, Kinect gaming was inaccessible to her because she lacked the ability to stand.

Even when they could play other games, participants sometimes believed that play was better for others with different abilities. P4 talked about *Beat Saber* [18], a virtual reality rhythm exergame that he plays at home. In *Beat Saber*, players swing motion controllers to slice colorful cubes to the beat of the music as they move side-to-side and duck to avoid oncoming obstacles.

*Beat Saber* can provide vigorous exercise for standing players, but P4 uses a wheelchair. He described the way that he plays as inferior to how others play: “*Let’s say that someone who can stand and move around to some degree, it would certainly suit them a lot better than it would me... When you’re standing, again, you’ve gotta move your body, but you have to move your whole body rather than just, you know, moving your... just kinda tilting your head.*” Even though he can play *Beat Saber*, P4 believed that play would be better for those who can stand.

As a consequence of these prior negative experiences, adapted interfaces can make players feel disabled, even when they are able to play. P6 recognized that players who can pedal the viva2 were able to control aspects of play that he could not, which he found diminished his experience: “[Playing the games] relied on what you were putting into, kind of like, the [viva2] as well. That’s a component of it. So, not being able to change what the input into the [viva2] would be is just kind of the diminishing part of it.” (P6) He said that using the viva2 as part of the games’ hardware interface indicated to him that they were not designed for him. We asked P6 if he believed that he missed out in playing *Dino Dash* and *Dozo Quest*. Although he recognized that there were parts of the games that he could not control, P6 said that he preferred being able to play in a limited way over not being able to play at all.

### 6.3.2.2 Major Theme: *Automation Confusion*

We explained to participants how to play each game using the inputs under their control, but did not explicitly explain that an AI copilot would be controlling aspects

of the avatar's behaviour. This confused two participants who noticed that some of their avatars' activities were automated, making it more difficult for them to identify inputs under their control and learn how to play. In this section, we describe two minor themes related to the confusion participants experienced due to automation.

**Minor Theme: *Understanding Source of Avatar's Behaviour*** Of the four participants who played with partial automation, only those who had difficulty co-ordinating their control with the copilot noticed its effects. Participants who were aware of the copilot sometimes found it difficult to recognize how their inputs changed their avatar's behaviour. This theme explores how participants made sense of partial automation and how it affected their experiences.

When the copilot performs actions that players expect, players may not notice the copilot. Only two participants recognized that their avatars' activities were not fully under their control. When asked if he believed he was in control while playing *Dino Dash*, P4 replied: “*Yes, fully in control!*” P4 played using the gamepad with the games’ Movement input automated. P3 used the same hardware interface as P4 and when asked the same question he gave a similar answer. Neither participant indicated that they were aware they had less than full control of their avatar in both *Dino Dash* and *Dozo Quest*.

Players became aware of the copilot, however, when the actions of the copilot made the game more difficult to play. P5 noticed the copilot while playing *Dino Dash*. She played using the joystick and bite switch with Movement automated, which made her dino move forward constantly. She found this difficult to control, saying: “*It was always in motion and I'm only... You're only controlling like the direction that it goes in, and I was trying to control everything*

*about it.*" P5 did not experience this confusion while playing *Dozo Quest*. She believed that she could control the dozo's speed even though this was under the copilot's control. When asked if she had difficulty playing *Dozo Quest*, she said: "*No, I could control it all so it was great!*" P5 was only aware that the Movement input was automated when it made playing more difficult.

Since both the participants and their copilots controlled the avatar, partial automation made it unclear to the player what they could control. P6 played using the bite switch, with both Movement and Direction automated. He controlled only one of three inputs in *Dino Dash*, and was unsure which aspects of the dino's activities could be attributed to him. During the interview, he was uncertain if he was able to affect the dino's direction. He said: "*It was just hard to tell, when I was trying to move the dinosaur in the right direction, about how to do that properly and stuff*" P6 encountered similar confusion while playing *Dozo Quest*. He said that "*[Dozo Quest] was still another one that was, like, hard to figure out what was doing what, and how to kind of go about it.*" Although he was able to correctly identify the dozo's dash attack as an action under his control, he was unsure if there were more.

**Minor Theme: Learning to Cooperate with Partial Automation** When participants recognized that there were aspects of the games that they did not control, it made play more difficult. Even when the copilot's behaviour was predictable, participants had difficulty coordinating with it. One participant found it frustrating being unable to influence the copilot's control of inaccessible inputs. This theme illustrates how participants learned to cooperate with their AI copilots.

P5 understood what her AI copilot was doing, but found that even predictable AI can be difficult to play with. The two participants who were aware of the copilot had difficulty playing the game when the copilot did not do what they wanted. This was not a question of understanding the split of control (as in the previous section), but a problem of coordinating with a copilot that at times performed undesired actions. P5 said: *“I’m just not used to playing games like that. So, for me to wrap my head around it it took a little bit more time.”* Although playing with automation was confusing initially, P5 was able to figure it out. *“I was trying to wrap my head around trying to control it better. So, it was just kind of difficult for me to grasp that in a sense. But then once I did and kind of play around with it and realize how I could control it a little bit better then it was more fun.”* (P5) Even when players understand how automation affects their avatar’s activities, they may still have difficulty using this inert knowledge.

P6 did not understand what the copilot was doing, so he found that ambiguity makes sharing control frustrating. P6 was unsure which of his avatars’ activities were under his control. While playing *Dino Dash*, he tried to influence the dino’s direction by tilting the bite switch. *“I think [it was frustrating] just not fully understanding the physical trick to the game. Like how to do things.”* (P6) His confusion about the games’ control responsibilities increased the time for him to learn to play effectively: *“[The frustration] wasn’t anything major. It just was something that I figured out and moved on from.”* (P6) Despite his

difficulty, P6 believed he could have more control over the game with more experience. Players who cannot recognize how their actions affected the game's outcome may become frustrated when that outcome is not favorable.

The accessibility approaches used in our study enabled participants with vastly different abilities to play the same games using controls personalized to their abilities. They enjoyed leveraging their expertise using familiar devices, but also believed that play may be better for others with abilities they lacked. Partial automation made the games accessible to participants with more profound motor impairments by providing inputs that they could not control themselves. However, this made it more difficult for some participants to recognize how their actions affected the avatar's behaviour and learn to cooperate with their AI copilot.

## 6.4 Design Implications for Partial Automation

We have shown how partial automation can enable players with very different physical abilities to play the same games. As presented in section 6.2, all participants in our study were able to engage in all important aspects of playing the games, answering our research question of whether partial automation makes *Dino Dash* and *Dozo Quest* more accessible. Our analysis of participants' interviews indicates that participants liked the increased accessibility and personalization that partial automation affords. Participants valued playing with devices that they found familiar and empowering. However, they said that being required to play the game differently from others could make them feel disabled. A minority of participants

experienced *automation confusion* that made understanding the copilot's behaviour and coordinating with it more difficult.

Our results indicate that rehabilitation games featuring partial automation may confer important benefits to patients in spinal cord rehabilitation. We will discuss these potential benefits in Chapter 9 and turn our focus now towards several insights into the design of partial automation that we discovered through our analysis. While it appears that partial automation can extend the accessibility afforded by universal design and interface adaptation, players may have difficulty understanding what the copilot does. When sharing control with partial automation, players may require further assistance in the form of awareness cues that help them to monitor and anticipate the copilot's behaviour.

#### 6.4.1 Broad Accessibility Through Combining Approaches

In Chapter 3, we presented a novel ontology of techniques for improving games accessibility. The results of our study illustrate how these techniques can be combined to make games more broadly accessible. While the *Dino Dash* and *Dozo Quest* games were created using universal design, they were unplayable by four of our six participants in their original form. To make the games playable to these four, personalization was required. Interface adaptation allowed two participants to control inaccessible inputs, and partial automation was used by four participants.

Alternative devices helped P5 and P6 to be able to play. P6 would not have been able to play at all had the bite switch not been available as an alternative to the standard game controller's face buttons. P5 was able to control both Direction and Action inputs only because of the availability of a larger joystick and the bite switch.

These participants' experiences show that alternative interfaces can help reduce the need for partial automation, allowing players to control inputs that would have been inaccessible with stock hardware.

Our thematic analysis shows that the availability of alternative interfaces goes beyond simply providing accessibility. Players valued being able to use controllers that they find familiar, and felt a sense of accomplishment in being able to make use of their prior knowledge (Section 6.3). For example, P5 noted that the joystick she used to play the games was similar to the joystick on her power wheelchair, which made her personalized hardware interface immediately familiar. This enabled her to transfer her wheelchair skills to playing the games. Similarly, P6 said that playing with his power wheelchair controls would be better than using an unfamiliar interface. While playing *Dino Dash*, he tried to control the dino's direction by tilting his head, similarly to how he controls the movement of his power wheelchair. When possible, therefore, games should allow players to use their own devices. Familiarity enables players to leverage their experiences of using other devices in daily life, possibly enhancing players sense of accomplishment.

#### 6.4.1.1 Automation Confusion

While partial automation combined with other approaches can broaden a game's accessibility, sharing control may also make players' control of accessible inputs more difficult and reduce players' agency. Players using partial automation need to cooperate with an AI copilot that performs tasks that players may not understand or even be aware of, which may confuse them about how to perform their own tasks. To better explain automation confusion, we refer to the game interaction model

of Yuan et al. [330] introduced in Chapter 2. Players first *receive stimuli* from the game, which they use to *determine responses*, and then *provide inputs*. These inputs generate new stimuli, triggering another iteration of the model’s steps.

The core goal of partial automation is to allow people to play games even when they cannot provide some of the necessary inputs. Using an AI copilot to provide the input can disturb the flow of players’ interactions with the game, as they see stimuli which may be a result of the copilot’s actions rather than their own. Player’s attempts to build an accurate mental model of the game may be hindered by the decoupling of clear cause and effect. More specifically, sharing control with an AI copilot introduces ambiguity into the normally tight correspondence between the inputs a player provides and the stimuli they receive.

For example, P6 tilted his head while using the bite switch as if to direct his dino in the direction he was tilting. His superstitious belief that he could influence the copilot’s behaviour caused him to misinterpret the stimuli he received – because the copilot’s actions were close enough to his desired actions, he incorrectly extended his mental model with the belief that tilting his head controlled the direction of his avatar’s movement. With only partial awareness of the inputs provided by the copilot, players may misinterpret the game’s stimuli in ways that made determining a response more difficult. P5’s difficulty coordinating with her dino indicates that partial automation can also make determining responses more difficult, even when players understand the game’s stimuli. She understood that her dino’s movement was driven by something else, but had difficulty determining how to coordinate with it. In these two distinct ways, players’ automation confusion may affect their abilities to understand what caused a particular stimulus or to determine how to

respond. If it is not addressed, players' confusion may lead to a disabling reduction in agency (e.g. P6's frustration trying to influence his dino's movement) due to disparities between their real and perceived control of the game.

#### 6.4.1.2 Diminished Agency and Experience

In the context of games, agency has been defined as players' perception that the actions they take (i.e., the inputs they provide) determine the game's outcome (i.e., the stimuli they receive) [252]. When players believe that they can control inputs that they cannot, such as when P6 tilted his head to direct his dino, their sense of agency may be diminished if the outcome is different than expected. This was the most common source of participants' automation confusion, although it did not always reduce players' feelings of agency. P3, P4, and P5 all believed that they were in full control of *Dozo Quest*, despite the Movement input being automated. This was not the case for P6 who became frustrated when his dino ignored his directions. To avoid such misunderstandings, games using partial automation may need to better support players' awareness of the copilot's actions.

Statements made by P5 and P6 indicate that the copilot itself may also affect players' feelings of agency. P5 recognized that her dino moved of its own volition and said that she would have preferred to control the Movement input herself. P6 recognized that he could not control Movement with the viva2, which he said diminished his experience of play.

Games can bolster players' agency by providing an *illusion of agency*—presenting inconsequential choices as meaningful [252]—however, deceiving players about

what inputs they control is easily detected whenever the copilot performs an action that the player did not expect. Instead, we propose that partially automated games improve their presentation of the copilot to the player by augmenting players' awareness of the copilot's actions and intentions.

#### 6.4.1.3 Communicating Awareness of Agent's Actions and Intentions

We left it up to participants to discover how to play using their personalized interface, which caused unnecessary confusion for P5 and P6. They were unsure of what the copilot could do, what it was doing from moment to moment, and why it was making the choices it did. This made it difficult for P5 and P6 to determine what they were responsible for controlling and to develop strategies around these responsibilities. Recent guidelines for human-AI interaction recommend that designers make clear what the copilot is able to do and communicate information relevant to the copilot's context to the player [10]. Fortunately, this sort of communication can work in many ways.

Cooperative communication mechanics, such as those described in Chapter 4, could be used by the copilot to share its current actions, intentions, and plans. Supporting mechanics could include emotes, gestures, or context sensitive messages. In *Dino Dash*, the copilot could highlight the dino or egg it is chasing and in *Dozo Quest* it could highlight the platform it is trying to jump onto. The resulting improved communication could enable players to both infer the cause of the copilot's current behaviour and predict its future behaviour.

Communication between humans engaging in cooperation has been extensively investigated in the context of *awareness widgets* for groupware [119; 122] and more

recently for games [28]. Stach et al. found that *information-rich embodiments*—awareness widgets that communicate salient information about game characters using glyphs—enabled players to develop better strategies in a *Spacewar!* [260] clone [281]. It may be that information-rich embodiments could also enable players using partial automation to better understand the copilot.

As explained by Gutwin et al., awareness cues correspond to one or more *elements of workspace awareness* [120; 121]. Personalized icons might indicate *identity*—“*who is in the workspace?*”—while color-coded carets indicate *authorship*—“*who is doing that?*”. Our participants’ feedback indicated that players sharing control with an AI copilot may have their own set of questions they need answered. They wanted to know what parts of the game they could control, what parts the copilot could control, and why the copilot was doing what it was doing. It may be possible to answer all of these questions, and any more that arise, using cooperative communication mechanics, information-rich embodiments, and other forms of awareness cues.

## 6.5 Summary

We have established partial automation’s efficacy in improving the accessibility of digital games to players with motor impairments and demonstrated its capacity to broaden the accessibility afforded by other games accessibility techniques, including universal design and interface adaptation. All of our study’s participants were able to play both *Dino Dash* and *Dozo Quest* with personalized assistance from partial automation. Their experiences of playing the partially automated study games and their reflections on how accessible gaming might have improved their rehabilitation

experiences indicate that playing games with partial automation can retain the fun of the original game.

While personalizable partial automation fulfilled its primary objective of making games more broadly accessible yet still fun, we identified a potential confound. Two participants had significant difficulty cooperating with and understanding the behaviour of their AI copilots. P5 wanted her avatar to stop but it kept moving while P6 wanted his avatar to move where he pointed and sometimes it did. Feedback from these participants, as well as the apparently superstitious head tilting exhibited by P6, indicate that partial automation can engender a sort of *automation confusion* that makes cooperating with the copilot more difficult. Although partial automation can excise disabling aspects of play, as per the equal partner pattern, it may also increase the ambiguity of the game's stimuli and lead players to develop bad mental models. In the next chapter, we propose a grounded theory to help designers of partially automated games to identify and address automation confusion.

# CHAPTER 7

---

## Automation Confusion

---

**W**HILE partial automation can improve a game's accessibility, it can also make players confused about how the game is controlled. We saw in the previous chapter how partially automated action games confused two of six players. They wanted their avatars to do specific things and became confused and frustrated when their avatars did things other than what they wanted. One player found cooperating with the copilot difficult because it would always move forward while her chosen strategy required it to stop. Another player thought that he could make his avatar do things that he could not and did not understand why his avatar would not do what he commanded it to do. We term this phenomenon *automation confusion* and will present a study in this chapter to elucidate how and why automation confusion occurs. We anticipate that better

understanding automation confusion will help developers to reduce its effects in the games they create.

If automation designed to simplify a game can inadvertently make players confused about how the game works, then designers need to understand players' confusion. As explained in Chapter 3, some commercial games have used automation as a form of player assistance and its use by designers unaware of automation confusion may have unforeseen consequences. Partial automation adds to what players need to understand; they need to understand the game's rules, which actions they can control, and which actions the copilot can control. In Chapter 4, we described known problems of human-machine cooperation and explained how automated systems have been designed to support users' awareness and understanding. However, it is not yet known how common or problematic automation confusion is in partially automated games, what types of confusion can occur, or how players become confused by the copilot.

This chapter contributes empirical evidence that automation confusion can occur and provides a theory of the types of confusion that arise when control of games is partially automated. To provide designers with knowledge about how to reduce automation confusion, we conducted a study in which ten non-gamer adults played two partially automated games and reported their understanding of the games during play. We chose to recruit non-gamers for this study because, as explained in Chapter 2, playing digital games can be overly challenging for non-gamers and we believed that partial automation may help them to play with their gamer friends and family. We also believed that non-gamers would be especially susceptible to automation confusion due to their lack of familiarity with other games. Indeed, we

found that participants' mental model errors led them to misattribute the causes of their avatars' actions. When participants were unable to make their avatars do what they wanted, they looked for alternative ways to control their avatars. Most participants believed that they could control some actions that they could not, while some others stopped believing that they could control the games at all.

The major contribution of this research is a *grounded theory* of automation confusion in non-gamers. A grounded theory is an explanatory framework that is derived from and therefore *grounded in* a collection of data related to some phenomenon of interest [106; 48; 40; 57]. It is helpful in making sense of phenomena with no existing theoretical explanation and provides mechanisms to ensure that the resulting theory is sufficiently general to explain new observations in the same substantive area. Any type of data or multiple types of data can be used to construct a grounded theory, but grounded theory methodology is often employed in the social sciences to construct theories based on qualitative data. For example, Glaser and Strauss famously used grounded theory methodology to discover a theory of how awareness that someone is dying affects social interactions related to that person [106], but one could also use grounded theory methodology to construct a theory of players' *emotional exploration* in games [56] or a framework of game mechanics [298]. In our case, we sought to explain how players' interactions with partially automated games might affect their mental models of how these games work. We believe that such a theory would enable game designers to identify the types and causes of automation confusion in their players, thereby giving designers the tools they need to reduce automation confusion's effects. In the next section, we will describe our study's design, the participants we recruited, and the partially

automated games they played. The primary research question addressed in this work is **RQ3**: “*What are the sources and types of confusion engendered by partial automation?*”

## 7.1 Study Design

To better understand whether partial automation can make non-gamers confused, we recruited ten non-gamer participants to play two novel partially automated games: *Ninja Showdown* and *Spelunky*. This study was approved by the General Research Ethics Board of Queen’s University (Appendix B). During play, participants were prompted to think aloud and discuss anything in the games that they found confusing. Gameplay logs and screen capture video with eye tracking were recorded to provide post-hoc insight into which buttons participants pressed and which objects on screen they looked at. After playing both games, participants were interviewed about their experiences, their opinions of the games’ avatars, and which avatar actions they believed that they could or could not control. This combination of gameplay, think-aloud, and interview data was analyzed using grounded theory methodology [106].

### 7.1.1 Partially Automated Games Used in the Study

Partial automation reduces the number of controls and mechanics that players need to master. It may help non-gamers, who have difficulty controlling and understanding games, to play with their gamer friends and family. To better understand non-gamers' experiences of partially automated games, and to characterize the confusion that can arise while playing them, we created two partially automated games. These games represent different genres (i.e., fighting & platformer) and use different forms of automation (i.e., input automation & one-switch), as described in Chapter 3.

Both games are controlled using only one button, the spacebar, although they afford different styles of play. The first is a fighting game called *Ninja Showdown* in which the player controls one of their avatar's attacks, while its other attacks are automated (i.e., input automation). The second is a recreation of an existing platformer called *Spelunky*, in which the player can make their avatar do different actions using the same button, while the avatar's movement is automated (i.e., one-switch)

Both games were designed with guidance from literature in partial automation, human-machine cooperation, and awareness. We carefully designed these games so that non-gamers' automation confusion could be attributed to partial automation, rather than the games themselves. *Ninja Showdown* was designed to provide players sufficient time to choose an action and observe its outcome. It replaced a game called *Meaty* whose real-time actions confused some gamers and non-gamers. *Spelunky*, as a clone of an existing commercial game, was not found to confuse players.



Figure 7.1: Screenshots from *Ninja Showdown*'s tutorial. In the image on the top left, Emi turns her head to look at the player, indicating that she is awaiting player input. In the bottom left, she displays an Intention cue indicating that she intends to use a Bomb if the player does not tell her to use the Sword. In the image on the bottom right, Emi has initiated the Bomb attack and displays an Action cue, indicating that she decided to use the Bomb.

To help induce functional mental models of the partial automation in players, we created tutorials that teach players which actions they can control and how to control them (Figures 7.1 & 7.2). These tutorials were designed iteratively through multiple rounds of usability testing with gamer and non-gamer subjects. To help players make sense of their avatar's automated actions, we created awareness cues informing players of its current actions and intentions (Table 7.1). These cues were also designed and tested iteratively, as test subjects were quizzed on their meanings. In this section, we describe the games used in this study and explain how we designed their automation.

### 7.1.2 *Ninja Showdown*

*Ninja Showdown* is a simple fighting game that was inspired by *Rock, Paper, Scissors* and designed for use in this study. The player controls a ninja avatar, named Emi, and is tasked with defeating a ninja opponent, named Takeshi. A game of *Ninja Showdown* lasts three rounds and each ninja can do one attack each round. A ninja can do a **Sword** attack, a **Bomb** attack, or a **Dart** attack. Each attack beats another: **Swords beat Bombs, Bombs beat Darts, and Darts beat Swords**. When a round begins, the announcer counts down from three and when the countdown reaches zero both ninjas do their chosen attack. On the count of two, the player's opponent chooses his weapon and displays it for the player to see. Once a ninja chooses an attack, their choice cannot be changed. If a ninja chooses the weapon that beats their opponent's, then that ninja gets a point and their opponent loses a point. If both ninjas do the same attack, then the attacks cancel each other out and no points are gained or lost. A full playthrough comprises 10 games, for a total of 30 rounds. *Ninja Showdown* was designed to be simple to facilitate analysis of automation confusion, without the confound of confusion arising from complex game mechanics.

**Players can command Emi to do the Sword attack by pressing the spacebar. Otherwise, the copilot makes Emi do either the Bomb or the Dart attack.** A tutorial tells players that Emi will choose an attack on her own if the player does not command her to use the Sword. In this way, *Ninja Showdown* implements input automation, as it delegates control of a subset of the game's inputs to the copilot. To help players recognize and anticipate automated actions, *Intention* cues, indicating which move Emi intends to do, and *Action* cues, indicating which move

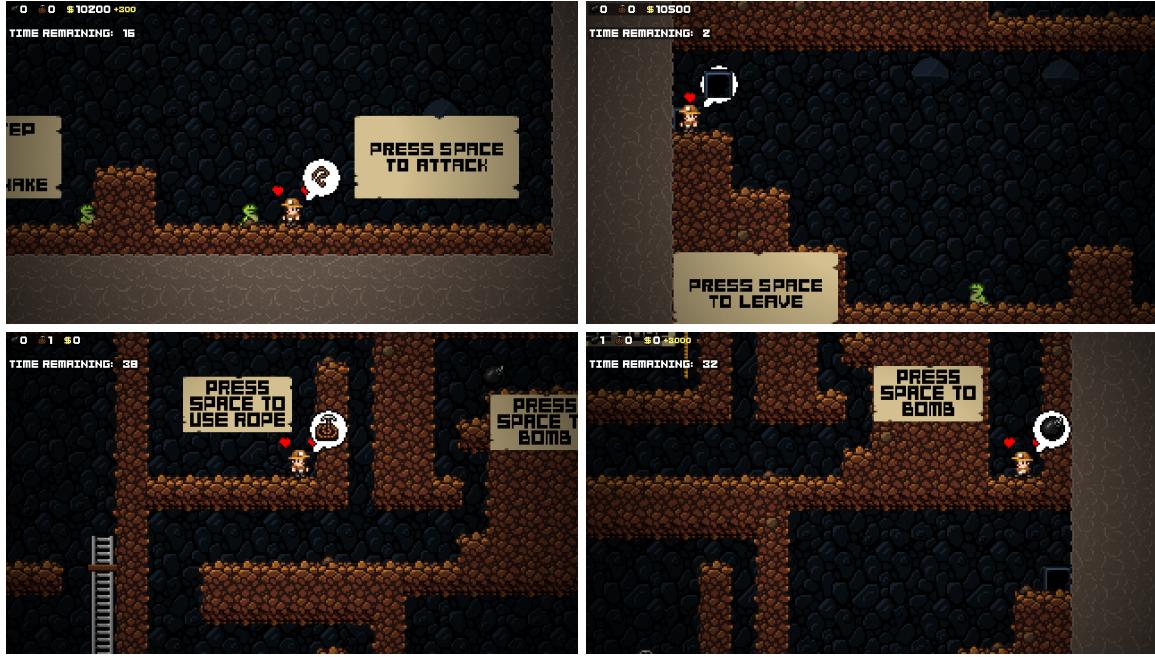


Figure 7.2: Screenshots from *Spelunky*'s tutorial. In each image, an explanation for how to make the avatar perform an action, as well as the action's corresponding Option cue, are displayed. Refusal cues were purely auditory and so are not visualized.

Emi has chosen to do, are presented to the player (Figure 7.1 & Table 7.1). Emi's intentions are represented by a thought bubble beside her and Emi's chosen actions are represented by a speech bubble above her head. These cues were designed to inform players of what the copilot will do in future and enable them to correctly attribute automated actions to the copilot.

### 7.1.3 *Spelunky*

*Spelunky* is a recreation of the popular game of the same name [209]; it was developed by Øyvind Strømsvik<sup>1</sup> and partially automated for use in this study. The

<sup>1</sup>Original source files available at <https://github.com/oyvind-stromsvik/spelunky>

player controls an explorer avatar that runs, jumps, climbs, and smashes its way through labyrinthine caves in search of loot and a way to escape. If the player's path is blocked by a thick wall, the avatar can deploy a **rope** to climb over it. If the player's path is blocked by a thin wall, then the avatar can throw a **bomb** to clear a path through it. *Spelunky* levels contain dangerous snakes, spiders, bats, and cavemen that the player can fight using a **whip** attack. If an enemy touches the player's avatar, then the avatar loses one of its two health points, represented by hearts floating above its head. Players are allotted between 30 and 120 seconds to complete each level.

**Players can command their avatar to use the whip, the rope, and the bomb by pressing the spacebar. The copilot decides which action occurs, if any, when the spacebar is pressed.** In this way, *Spelunky* implements one-switch automation by reducing players' interactions to a single button used to control multiple actions. This reduces the number of decisions players need to make; players decide *when* to do an action, and let the copilot decide *what* action to do. To help players understand and anticipate their *Spelunky* avatar's actions, *Option* cues display the avatar's selected action and *Refusal* cues inform the player that their spacebar press did not result in an action (Figure 7.2 & Table 7.1). Whenever the avatar is in range to hit an enemy, needs a rope to climb up, or can clear a path with a bomb, a thought bubble containing the needed item is shown next to the avatar. The player then has the option of making the avatar use that item by pressing the spacebar. Should the player press the spacebar when the copilot has not selected an action, voice recordings of an actor saying “Nah” or “Huh” are played. **The avatar's walking,**

Table 7.1: Each game's awareness cues and their meanings.

Game	Awareness Cue	Meaning
<i>Ninja Showdown</i>	Action	<i>What</i> automated action the avatar is currently performing
	Intention	<i>What</i> automated action the avatar will perform
<i>Spelunky</i>	Refusal	No action was performed <i>when</i> the player pressed the button
	Option	<i>What</i> action the avatar will perform <i>when</i> the player presses the button

**running, jumping, and climbing are controlled exclusively by the copilot**, so players do not need to learn how to control these actions.

#### 7.1.4 Summary

The games were designed to be simple to play and easy to understand. The tutorials were designed to tell players explicitly which actions they can control and how they can control these actions. The awareness cues were designed to inform players of what the copilot is doing and what it will do. Both games provide players all of the information they need to learn how the games work, how they are controlled, and how to cooperate with the copilot. Both games successfully reduced the number of inputs players needed to control, which consequently reduced the number of decisions they needed to make.

This does not always come without a cost. the copilot in *Ninja Showdown* reduces the complexity of players' control, while slightly increasing the complexity of players' decision-making. When the player can choose any of the weapons, *Ninja Showdown* is no more complicated than *Rock, Paper, Scissors*. But, when some of the game's inputs are automated, some of this choice is delegated to the copilot. Players need to determine whether it is possible to win the round, which is always possible in the manual version. In contrast, *Spelunky* simplifies both how players control the game and the decisions players make.

Table 7.2: Participants' demographic data including the game genres they had played before (Genres), the number of hours they spent playing games each week (Current Gaming H/W), the number of hours they spent playing games each week at their peak (Peak Gaming H/W), whether they mentioned wanting to play games with friends or siblings (Friend), and whether they mentioned wanting to play games with their children (Parent). Participants' mean age was 38.8 and their mode gender was woman.

ID	Age	Gender	Genres	Current Gaming H/W	Peak Gaming H/W	Friend	Parent
P1	25	Man	Platformer	0	0	✓	
P2	31	Woman	None	0	0	✓	
P3	47	Woman	None	0	0	✓	
P4	53	Man	Puzzle, Sports	0	2	✓	
P5	23	Woman	Fighting, Platformer, Simulation	0	3	✓	
P6	28	Woman	Adventure, FPS, RPG, Strategy	0	3	✓	
P7	26	Woman	Puzzle	0	2	✓	
P8	57	Woman	Puzzle, Rhythm	0	1	✓	
P9	47	Woman	Puzzle	0	0	✓	
P10	51	Woman	Puzzle	1	2	✓	

### 7.1.5 Participants

Non-gamer participants were chosen because they may benefit from the simplification afforded by partially automated games. We wanted to understand how these games are experienced by players who may have difficulty playing games without partial automation. A recruitment poster was posted on local social media pages and circulated within several departments at Queen's University. Participants were required to have played fewer than 100 hours of video games in their lives, be aged eighteen or older, be able to use a keyboard, and speak English. Since colloquial use of the term “video game” is ambiguous, we did not recruit respondents who identified as gamers or whose responses indicated that they had ever played action games regularly. Participants took part in lab-based study sessions, lasting approximately an hour and a half, and were given a \$20 honorarium.

### 7.1.6 Procedure

Before each study session, participants provided their informed consent to participate. They were seated at a desk with a 31.5" Benq EW3270U monitor, a full keyboard, and a Tobii Eye Tracker 5. Participants completed a short demographic questionnaire asking about their age, gender, and familiarity with popular gaming genres. They were then interviewed about the people in their lives that play video games and their attitudes towards playing with these people. Following the first interview, the researcher explained how participants should think aloud while playing the games and informed them that the researcher would periodically ask them questions.

Before playing either of the digital games, participants first engaged in a simple drilling exercise with cards to familiarize them with the *Ninja Showdown* scoring rules. Informal testing during the *Ninja Showdown*'s development indicated that players had difficulty remembering which weapons beat each other, so this activity was intended to teach participants how to win before teaching them how to play with the copilot.

Before playing each game, the researcher read to participants a short primer explaining the game's goals, which actions they could make the avatar do by pressing the spacebar, and that their avatar was "pretty smart" and could do other actions on its own. During the explanations, the games' visual elements, such as avatars and enemies, were identified in screenshots to help participants recognize them during play. Both games began with a short tutorial section. In the *Ninja Showdown* tutorial, automated avatar actions and the meaning of the awareness cues are explained during scripted gameplay sequences with explanatory text (Figure 7.1).

In the *Spelunky* tutorial, large signs in the game's first two levels instruct players to press the spacebar to make the avatar attack, use ropes, and throw bombs (Figure 7.2). Participants played *Ninja Showdown* first and *Spelunky* second, each for approximately 15 minutes.

While playing the games, participants were encouraged to think-aloud and were prompted to speak using questions such as “*What are you thinking?*”, “*Why did you do that?*”, and “*How do you know that?*” Screen capture video, including an overlay displaying where on the screen participants were looking, and audio of participants speaking were recorded using OBS<sup>2</sup>. Both games recorded frame-by-frame gameplay data, including all of the keyboard keys participants pressed and held as well as the visual elements they looked at. After playing both games, participants were interviewed for approximately 30 minutes about their experiences, their understanding of how the games were controlled, and their opinions of the avatars. Participants were asked whether they ever believed that they could make their avatars perform specific actions in the games, including actions that players could not control. The interviewer also asked whether participants ever wanted to or tried to make their avatar perform actions that they could not make them do and encouraged participants to explain their thinking during these incidents. Participants' think-aloud data and explanations of their thinking during the final interview were used to elicit their mental models and determine how output from the games made them confused about how they were controlled.

Table 7.3: Definitions for all of automation confusion's concepts.

<b>Theory of Automation Confusion</b>		
<b>Types of Mental Model Errors</b>		
Category	Concept	Definition
False Causation	Over-Attribution	Player believes that they control actions that they do not
	Under-Attribution	Player believes that they do not control actions that they do
Explanation Errors	Extra-Rule	Player has a rule that does not explain any aspect of the game's output
	Overly-Simple-Rule	Player has a rule too simple to explain some aspect of the game's output
	No-Rule	Player has no rule to explain some aspect of the game's output
<b>Attitudes Towards the Games</b>		
Category	Concept	Definition
Analytical	Uncritical	Player plays without trying to improve their understanding
	Critical	Player plays to improve their understanding
Emotional	Frustrated	Player feels frustrated by avatar's actions
	Uninvolved	Player feels uninvolved in play
<b>Behaviours Resulting from Confusion</b>		
Category	Concept	Definition
Learning	Exploration	Player presses buttons to discover their effects
	Confirmation	Player presses buttons to confirm that an expected effect occurs
	Contemplation	Player observes the game's output without pressing buttons
Superstitious	Shadowing	Player presses buttons along with automated actions
	Mashing	Player presses buttons with no intended effect
	Manner Modification	Player modifies the way they press buttons to change their effects
<b>Sources of Confusion</b>		
Category	Concept	Definition
Feedback	Misinterpreted Feedback	Feedback whose meaning players misunderstand
	Missed Feedback	Feedback that players do not notice
	Missing Feedback	Feedback that does not exist but might otherwise prevent confusion
Wrong Expectations	Inherited Expectations	Expectation that the game works like something else
	Incorrect Mappings	Expectation that buttons control actions because of arbitrary associations
	Wishful Thinking	Expectation of control borne of a desire for control

## 7.2 Results

In total, ten participants enrolled in our study. All participants answered “No” to the screening questions: “*Do you play video games?*”, “*Do you identify as a gamer?*”, and “*Have you played more than 100 hours of video games in your life?*”. All participants, except P10, reported spending zero hours each week playing video games. P2-4 and P8-10 were parents who wanted to play with their gamer children, but had difficulty controlling and understanding games. P1 and P5-7 mentioned feeling excluded from gaming activities because they were not as skilled as their gamer friends or siblings. Participants’ demographic and gaming data (Table 7.2) suggest that they were people for whom gaming is not, and has never been, a significant part of their personal or social lives.

All ten participants were able to play both games; nine of the participants experienced at least one episode of confusion about how at least one of the games was controlled. Participants’ gameplay, think-aloud, and interview data were analyzed using classical grounded theory methodology [106] from a constructivist perspective [48], as is typical of recent usage [40; 57]. The resulting theory is organized around the core category of *automation confusion*, which we organized into four properties: *types*, *attitudes*, *behaviours*, and *sources* (Table 7.3). Our analytical process is visualized in Figure 7.3 and described in the text below.

**Data Collection:** During each session, we documented participants’ behaviours and utterances that we believed held theoretical significance. During interviews, we asked participants to explain these behaviours or utterances.

---

<sup>2</sup><https://obsproject.com/>

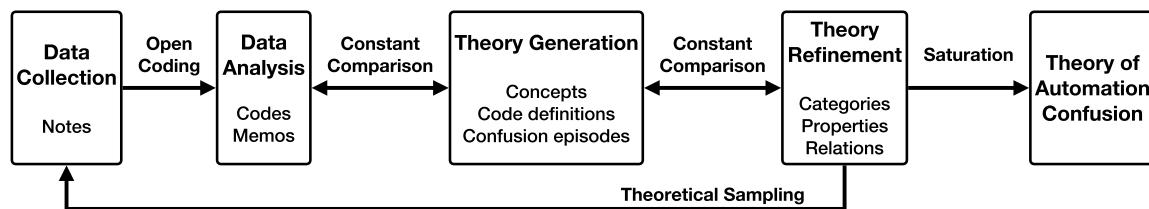


Figure 7.3: The analytical process we followed during theoretical development.

**Data Analysis:** After each session, the data were analyzed in isolation, yielding new codes (i.e., labels) and memos (i.e., free-form notes). There were often commonalities between participants' data, but the purpose of this initial analysis was to elucidate and explain their differences.

**Theory Generation:** During theory generation, the new data, codes, and memos were compared to our developing theory and used to detect poor fit in its concepts. When we were unsure whether the same concept explained two or more different observations, or when the same observation was explained by multiple concepts, we created short stories called confusion episodes (such as in Figures 7.4-7.6). By creating narratives from multiple slices of data, we ensured that our stories, and thereby our theory, made sense.

**Theory Refinement:** Comparing the codes, memos, and confusion episodes generated from different participants' data helped us to refine our concepts and led to the development of progressively abstract conceptual categories whose relations were apparent in the confusion episodes.

**Saturation:** When new data prompted the construction of new codes and concepts, we sought more data for further development. When new data prompted theoretical refinement, but not generation (i.e., P8), we concluded that we

had reached saturation. Significant theoretical generation occurred following P2, P5, and P7's sessions. Our developing theory was tested for fit (i.e., fit to the data) and grab (i.e., explanatory power) with P9 and P10, both of whom found the theory comprehensible and relevant to their own experiences of playing the games.

### 7.2.1 Automation Confusion

*Automation confusion* is a theory we developed of how players become confused by partially automated games. When both the human player and the copilot control actions in the game, players may become confused about which actions they can control. Our theory posits that players construct mental models of a game's rules by providing input to the game, interpreting the game's outputs, and then comparing their interpretations with their expectations. However, automated actions may lead to outputs that cause the player to learn incorrect rules or doubt correct rules; they may misinterpret what happens in the game and therefore learn an incorrect mental model of the game. Players may infer causal relations that do not exist and only appear coincidentally. As we will show, many aspects of a game's output can make players confused about how the game works. But perhaps most detrimental to their understanding is players' desire for control over the game; players want to make their avatars perform specific actions and can become frustrated when they cannot make them do those actions. As concrete examples, we now illustrate through storyboards how the games confused P2, P7, and P3. To disambiguate the Bomb in *Ninja Showdown* and the bomb in *Spelunky*, *Ninja Showdown* actions are capitalized and *Spelunky* actions are not.



Figure 7.4: Storyboard depicting P2 pressing the 'b' key to make Emi use the Bomb.

**P2:** P2 incorrectly believed that the 'b' key made Emi use the Bomb (Figure 7.4).

To test her hypothesis, P2 pressed the 'b' key to verify that it did what she expected. When Emi coincidentally pulled out the Bomb, P2 misinterpreted Emi's automated action as confirming that she made Emi use the Bomb.

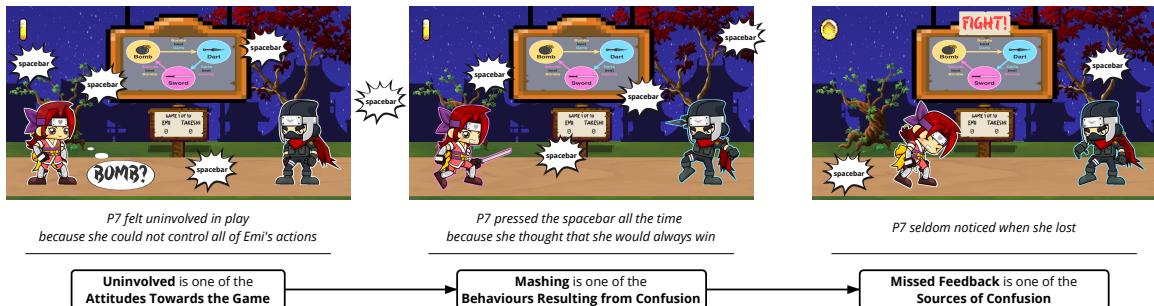


Figure 7.5: Storyboard depicting P7 mashing the spacebar because she felt unininvolved in play.

**P7:** P7 felt unininvolved in play because the avatars would do actions on their own (Figure 7.5). Being unable to control all of her avatars' actions frustrated P7 so much that she stopped trying to learn how the games worked and stopped paying attention to what happened in them. In *Ninja Showdown*, P7 pressed the spacebar incessantly, believing it would always cause her to win, and she often did not notice when she lost.

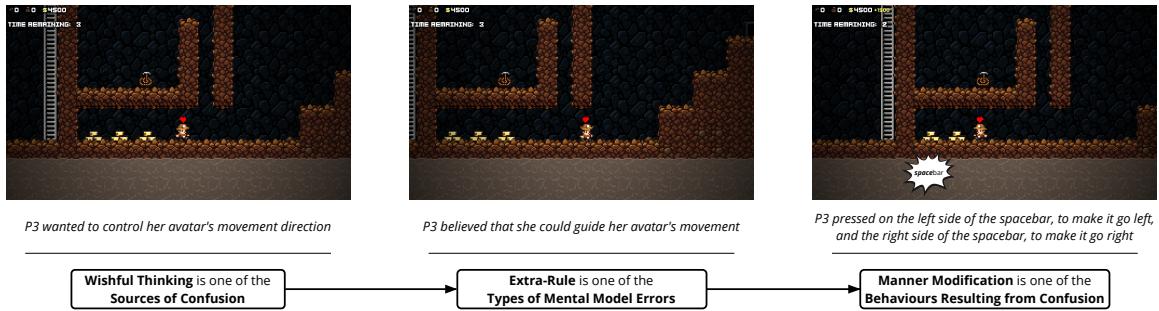


Figure 7.6: Storyboard depicting P3 pressing on the left and right sides of the spacebar to guide her avatar.

**P3:** P3 was often frustrated because her *Spelunky* avatar would move in directions that she did not want it to go (Figure 7.6). She wanted to influence its movement direction and took to pressing on different sides of the spacebar to signal her desires to the avatar. While playing, P3 explained that “*In my mind, I’m feeling if I [press on the left] it will go left and if I [press on the right] it will go right.*” P3 pressed on different sides of the spacebar to guide her avatar’s movement and sometimes it did what she wanted.

We found that participants exhibited a variety of **mental model errors** that influenced their **attitudes towards the games** and their **behaviours while playing them**. Participants’ behaviours determined the games’ outputs, which sometimes helped them to make sense of what was going on and sometimes created **sources of confusion**. In P2’s example, her incorrect mental model caused her to press the ‘b’ key, which indicated that she could control an action that she could not. In P7’s example, her feelings about the games caused her to press the spacebar with no specific intention and miss feedback that might have told her what her pressing did. In P3’s example, her desire to control her avatar’s movement led her to believe that she could by pressing on different sides of the spacebar. In this section, we

Table 7.4: Definitions for the types of mental model errors experienced by participants.

Types of Mental Model Errors		
Category	Concept	Definition
False Causation	Over-Attribution	Player believes that they control actions that they do not
	Under-Attribution	Player believes that they do not control actions that they do
Explanation Errors	Extra-Rule	Player has a rule that does not explain any aspect of the game's output
	Overly-Simple-Rule	Player has a rule too simple to explain some aspect of the game's output
	No-Rule	Player has no rule to explain some aspect of the game's output

explain our theory's four properties and describe how they relate to each other in ways that can affect players' confusion.

### 7.2.2 Types of Mental Model Errors

All participants, save for P4, constructed mental models that contained errors. While playing the games, participants were prompted to explain the games' rules as they understood them. When participants mentioned pressing buttons other than the spacebar, they were asked what they believed the buttons made their avatar do. As shown in Table 7.4, we categorize participants' mental model errors as errors of *false causation* and *explanation errors*. Errors of false causation occurred when participants misunderstood what caused the avatar to perform actions and *explanation errors* occurred when participants learned rules that incorrectly explained what happened in the games.

#### 7.2.2.1 False Causation: *Over-attribution & Under-attribution*

False causation errors occurred when participants misattributed the causes of avatar actions. Participants sometimes misattributed automated actions to themselves

(i.e., over-attribution) or misattributed their actions to the copilot (i.e., under-attribution). For example, P1, P3, P5, P7, and P9 all believed at some point that they could make their *Spelunky* avatars jump, an action that only the copilot could control.

**Over-attribution:** Over-attribution errors occurred when participants believed that they could control actions that they could not. P3 incorrectly believed that the spacebar made Emi use the Bomb, and not the Sword, in *Ninja Showdown*. When asked why she pressed the spacebar, causing her to tie instead of win the round, P3 sighed and said “*I wanted Bomb for the Dart... but the things was not there.*” P3 believed that she could control actions that she could not and became surprised when her avatar did not do them.

**Under-attribution:** Under-attribution errors occurred when participants did not believe that they could control actions that they could. P3 expected her avatar to use the Bomb attack when she pressed the spacebar, but was surprised when the avatar used the Sword instead. The avatar appeared to be ignoring P3’s commands, which caused her to question whether she could control either character in the game. When asked which character she could control, P3 said “*I don’t think I am able to control any of them, because there is no option for me to tell them what to use... [my avatar] must have the Sword to destroy the Bomb. So, where I can give him the Sword?*” P3 was unable to make her avatar do the actions she wanted and so stopped believing that she could control any action.

### 7.2.2.2 Explanation Errors: *No-Rule*, *Overly-Simple-Rule*, & *Extra-Rule*

Explanation errors occurred when participants believed in rules that incorrectly explained the game's output. The causal connections between players' inputs and the game's outputs can be thought of as 'explanation rules' that explain what is going on in the game. For example, P2, P7, and P10 all thought that they needed to press the spacebar, twice sometimes (i.e., P7 & P10), to make bombs explode in *Spelunky*; this was not an avatar action and neither the copilot nor participants could control it. Participants sometimes invented rules that did not hold up, came up with rules that were too simple to account for the variety of outcomes they observed, or failed to discover rules explaining part of the game's output.

**Extra-Rule:** Extra-rule errors occurred when participants believed in rules that did not explain any aspect of the game's output (e.g., *Takeshi does what Emi predicts*). P2, P3, P5, P7, and P10 spent some of their time playing *Ninja Showdown* believing that their avatar's Intention cues, which indicated which attack their avatar intended to do, actually predicted which attack their opponent would do. These participants would not look at their opponent at all, decide what to do based on their avatar's Intention cue alone, and then become surprised when they lost the point they thought they had won. P2 made her avatar use the Sword when she thought that her opponent would use the Bomb, even though he was already holding the Dart. When she lost the round, P2 laughed and said "*This is crazy! I think I do not understand the rules to play it.*" Participants who believed in rules that did not exist had difficulty making sense of the feedback they received.

**Overly-Simple-Rule:** Overly-simple-rule errors occurred when participants came up with rules that did not account for all of the different outcomes they observed (e.g., *spacebar makes the avatar do something*). While playing *Spelunky*, P1 understood that pressing the spacebar would make his avatar attack, use a rope, or throw a bomb, but he did not know which specific action his avatar would do. P1 also believed that pressing the spacebar would make his avatar jump, and he often wanted it to jump. He pressed the spacebar so prodigiously that he seldom had time to see his avatar's Option cue before it performed the action. When asked whether he knew what his avatar would do when he pressed the spacebar, P1 said "*I know what he's doing.*" P1 was content with the actions his avatar was doing and so did not try to discern which action it would do in particular.

**No-Rule:** No-rule errors occurred when participants did not have a rule to explain some aspect of the game's output (e.g., not knowing that the avatar can use ropes and bombs in *Spelunky*). P2 knew only that she could make the *Spelunky* avatar attack and was surprised when it did anything else. Any time her avatar threw a bomb, P2 became worried because she thought that it would hurt her avatar. In the game's third level, P2 pressed the spacebar to attack a nearby enemy while her avatar was preparing to throw a bomb. When she saw the bomb, P2 said "No. No. No. No. No. No. No. No. *I'm not in control of this avatar... I pressed the spacebar to kill the snake.*" She could not make sense of the Option cues, informing her of which action her avatar would do, and did not understand why her avatar would sometimes ignore her commands to attack.

Table 7.5: Definitions for the attitudes towards the games experienced by participants.

Attitudes Towards the Games		
Category	Concept	Definition
Analytical	Uncritical	Player plays without trying to improve their understanding
	Critical	Player plays to improve their understanding
Emotional	Frustrated	Player feels frustrated by avatar's actions
	Uninvolved	Player feels uninvolved in play

Participants were content with believing that they could control actions they could not. P1 and P5 wanted the *Spelunky* avatar to jump and liked that it did—coincidentally—when they pressed the spacebar. However, some participants became frustrated when they could not make their avatars do what they wanted or when the games played out differently than they expected. Participants' erroneous beliefs about how the games worked affected both their attitudes towards the games and their behaviours while playing them. Realizing that they misunderstood the rules of play prompted some participants to more critically analyze the game's output and press buttons in service of improving their understanding.

### 7.2.3 Attitudes Towards the Games

Participants' attitudes towards the games affected how they played with partial automation. Participants became frustrated when they did not like what their avatar was doing and could not make it do what they wanted. Frustration caused some participants to pay more attention and make sense of what happened. In some cases, participants became so frustrated that they felt uninvolved in play and stopped trying to understand what was going on. In this section, we describe the *analytical* and *emotional* attitudes that participants' confusion engendered (Table 7.5).

### 7.2.3.1 Analytical Attitudes: *Critical & Uncritical*

Participants adopted different analytical attitudes towards making sense of the games' outputs. When they were content with their understanding, participants played the games with an *uncritical* attitude, meaning that they were not trying to improve their understanding. However, when they could not explain their observations, some participants adopted a more *critical* attitude, meaning that they carefully observed the games' outputs to improve their mental models.

**Uncritical:** Participants were uncritical of the games when they were not trying to improve their understanding. P7 disliked *Ninja Showdown* so she did not care to improve her understanding of the game. P7 lost many rounds, but nevertheless concluded that all she needed to do to win every round was to press the spacebar. She explained that “*It’s too boring. Like, if I kept just pressing the spacebar I’ll win in all the games, right?*” *I don’t even need to see what she’s predicting. You don’t even need to hear, just press.*” P7 was uncritical in her analysis of the game and did not notice or care when she lost.

**Critical:** Participants were critical in their analysis of the games when they played with the intention of improving their understanding. While her opponent was holding the Sword, P5 saw her *Ninja Showdown* avatar’s ‘Bomb?’ Intention cue and pressed the spacebar to use the Sword in response. Noticing her mistake, P5 said “*Oh, wait no. I chose the wrong option... ’Cause I thought that guy had the Bomb, so I had to choose the Sword.*” Over the next several rounds, P5 explained what her observations told her about how the game worked. She described how “*Takeshi brought out the Dart, [my avatar is] supposed to*

*bring the Bomb, but instead she brought the Dart... That was the point where I was like ‘is she saying for herself or the guy?’” She carefully and critically analyzed her avatar’s actions and explained that “Later in the round it kinda, like, made sense that they were her thoughts about herself not Takeshi.”*

#### 7.2.3.2 Emotional Attitudes: *Frustrated & Uninvolved*

Participants adopted different emotional attitudes towards playing the games. When they were unable to make their avatars do what they wanted, some participants became *frustrated*. Often, participants’ frustration made them more interested in discovering how the games worked and caused them to search for alternative ways to control their avatars. However, when participants determined that they could not control their avatar in a way that they liked, they felt *uninvolved* and gave up on trying to control it. Frustrated and uninvolved were not the only ways participants felt while playing; P1 and P5 had fun believing they could make the *Spelunky* avatar jump and P6 felt accomplished when she correctly timed her attacks to hit the snakes. Feeling frustrated and uninvolved contributed significantly to P1-3 and P6-9’s confusion and were therefore included as concepts in our theory.

**Frustrated:** Participants were frustrated by the games when they could not make their avatars do the actions they wanted them to do. P6 did not know what to do when her opponent chose the Sword and became frustrated that she was unable to “rescue Emi” when her avatar intended to choose a Bomb. She wanted some action she could do to help her avatar and suggested that she should be made able to ‘reject’ her avatar’s selection with the ‘r’ key. But eventually P6 stopped being frustrated and realized that she could reject her

avatar's suggestions by choosing the Sword. She explained that "*When I stopped thinking about how to use the other weapons and what to press to have the other weapons, I tried to analyze how it works.*" Frustration caused by the avatar not doing what participants wanted caused some participants to more critically analyze the game's outputs.

**Uninvolved:** Participants felt uninvolved in play when they could not make their avatars do what they wanted. P7 felt uninvolved in playing *Spelunky* because her avatar made all the decisions for her. She said "*He's the one who's knowing how to get out of this maze. Like, I feel really like I'm not doing anything. I'm just doing what he wants.*" Just as in *Ninja Showdown*, P7 concluded that the optimal way to play *Spelunky* was to press the spacebar as often as possible. Since her avatar was "*Just using what he wants*", P7 did not need to improve her understanding of the game.

In sum, participants' emotional attitudes towards the games affected their analytical attitudes and consequently their behaviours. Frustrated participants pressed buttons and reflected on their observations to verify that their mental models were correct. Overly frustrated participants (i.e., P2, P3, & P7) stopped taking actions and stopped trying to make sense of their observations.

#### 7.2.4 Behaviours Resulting from Confusion

Participants cited many reason for pressing or not pressing buttons. Their erroneous mental models and attitudes towards the games caused them to exhibit both *learning* and *superstitious* behaviours (Table 7.6). Sometimes participants would press

Table 7.6: Definitions for the behaviours resulting from confusion exhibited by participants.

Behaviours Resulting from Confusion		
Category	Concept	Definition
Learning	Exploration	Player presses buttons to discover their effects
	Confirmation	Player presses buttons to confirm that an expected effect occurs
	Contemplation	Player observes the game's output without pressing buttons
Superstitious	Shadowing	Player presses buttons along with automated actions
	Mashing	Player presses buttons with no intended effect
	Manner Modification	Player modifies the way they press buttons to change their effects

buttons, or selectively not press buttons, to learn what would happen. Other times, participants superstitiously pressed buttons that had no effect, pressed buttons with no express intention, or changed the way they pressed buttons to make the avatars do different actions.

#### 7.2.4.1 Learning Behaviours: *Exploration, Confirmation, & Contemplation*

Participants exhibited learning behaviours when they selectively pressed buttons to learn their effects. Participants used three types of behaviours to improve their understanding of the games. They employed *exploration* when they pressed buttons to observe their effects, *confirmation* when they pressed buttons to confirm their effects, and *contemplation* when they did not press buttons to observe what would happen.

**Exploration:** Participants exhibited exploration when they pressed buttons to discover their effects. Participants often wanted to control actions that they could not and would explore the keyboard for ways to make their avatars do them. P1 knew that the spacebar made his *Ninja Showdown* avatar use the Sword,

but wanted to make his avatar use the Bomb and Dart attacks and tried pressing keys other than the spacebar to make it do them. When asked why he was pressing the arrow keys, P1 explained that “*I’m going to get to see what it does, because it’s only Swords*” Eventually, through a series of coincidences, P1 discovered that he could make his avatar use the Bomb with the left arrow key and the Dart with the right arrow key, even though he actually could not. Participants who went looking for the keys that made their avatars do what they wanted (i.e., P1-3, P8, & P9) often believed that they could control these actions.

**Confirmation:** Participants exhibited confirmation when they pressed buttons to confirm that they did what participants expected. P9 was frustrated that “[*Ninja Showdown*] doesn’t give you a chance to defend yourself” and tried pressing the ‘control’ key to make her avatar choose the winning weapon. Coincidentally, the avatar did what P9 wanted, which led her to suspect that she was the cause of the avatar’s action. At the beginning of the next round, P9 said “*I’m testing a theory*” and tried pressing the ‘control’ key again to confirm her suspicion. When asked why she believed that ‘control’ may make her avatar choose the winning weapon, P9 said “*Cause it just worked when I did it.*” When they doubted their understanding of how to control the avatars, P1-3 and P7-10 tried pressing buttons to confirm their effects.

**Contemplation:** Participants exhibited contemplation when they chose not to press buttons to observe what would happen. When asked whether she thought that she could make her *Spelunky* avatar jump, P8 said “*I don’t think so. I’m not sure. But, I think that if I were to play it again, I’d just watch what he*

*does more and pay more attention to what I actually have control over. [I was watching] for whether my pressing the bar had an action; for whether he would do things on his own without. So, just for the, you know, whether my pressing things was needed and whether it had an effect.*" While pressing buttons to explore and confirm their effects led some participants to develop superstitions, not pressing buttons and contemplating what happened helped participants to dispel them.

#### 7.2.4.2 Superstitious Behaviours: *Shadowing, Mashing, & Manner Modification*

Participants exhibited superstitious behaviours when they pressed buttons that did not have the intended effects. *Shadowing* occurred when participants pressed buttons along with automated avatar actions. *Mashing* occurred when participants pressed buttons with no specific intention or to avoid an undesired outcome. *Manner modification* occurred when participants changed the way they pressed buttons to change their effects.

**Shadowing:** Participants exhibited shadowing when they pressed buttons to make their avatars do actions they were already doing. Most participants (i.e., P1, P3, P5-10) at some point believed that they could make their *Spelunky* avatar jump or pick up items by pressing the spacebar. P5 immediately started tapping the spacebar along with her avatar's jumping in the first level of *Spelunky*. When asked if she thought that she could make the avatar jump, P5 said "*Sometimes I did but sometimes I didn't... I felt that it was my instinct that I was making him do, but sometimes I didn't press the key and he was doing*

*it by himself.*” Jumping was an action that only the copilot could control but, since her avatar was already jumping, P5 thought that she was contributing.

**Mashing:** Participants exhibited mashing when they pressed buttons with no intended outcome or to avoid an undesired outcome. P9 was unsure of what she could make her *Spelunky* avatar do but pressed the spacebar anyway. When asked whether the avatar did what she expected, P9 said “*I don’t know, because I don’t know what I’m expecting him to do.*” We then asked her why she pressed the spacebar at all and she said “*Force of habit. I feel like I have to... ‘Cause if I don’t, I die. See?*” P9 had no expectation for what her avatar would do when she pressed the spacebar; she believed only that if she did not press it she would lose.

**Manner Modification:** Participants exhibited manner modification when they modified the way in which they pressed buttons to modify their effects. If a desired outcome did not occur, participants sometimes suspected that they pressed the button at the wrong time or in the wrong way. P3 pressed on different sides of the spacebar to influence her avatar’s movement. Although P3 recognized that “*spacebar is a spacebar*” and that pressing on different sides would not produce different effects, she never entirely gave up hope that she could guide her avatar. When asked if pressing on different sides of the spacebar ever made her avatar do what she wanted, P3 said “*Yeah, yeah. It does. I feel like it does... I wanted to go like this and sometimes it’s happen and sometimes it’s not. Is that something, like, there is a connection?*”

Table 7.7: Definitions for the sources of confusion experienced by participants.

Sources of Confusion		
Category	Concept	Definition
Feedback	Misinterpreted Feedback	Feedback whose meaning players misunderstand
	Missed Feedback	Feedback that players do not notice
	Missing Feedback	Feedback that does not exist but might otherwise prevent confusion
Wrong Expectations	Inherited Expectations	Expectation that the game works like something else
	Incorrect Mappings	Expectation that buttons control actions because of arbitrary associations
	Wishful Thinking	Expectation of control borne of a desire for control

The participants who developed the most accurate mental models of the games were those who struck a balance between pressing buttons and not pressing buttons. As explained by P8, “*As [my avatar] was doing his stuff I was trying to pay attention to what he was doing, but I was also trying to interact... It only meant that sometimes I was just, you know, hitting it, as opposed to really paying attention: ‘is it actually doing anything?’*” Participants wanted to make their avatars do what they wanted them to do and so would press buttons to feel more involved in their avatar’s activities. They would misinterpret the games’ outputs in favorable and convenient ways that aligned with their expectations and desires.

### 7.2.5 Sources of Confusion

Participants developed erroneous mental models due to both the feedback they received and their expectations for how the games worked. Each participant approached the games with their own set of expectations and so imagined that the games would work in different ways. Participants’ incorrect expectations about how the games worked caused them to misinterpret the games’ outputs. In this section, we describe how *feedback* and participants’ *wrong expectations* caused them to become confused about how the games were controlled (Table 7.7).

### 7.2.5.1 Feedback: *Misinterpreted Feedback, Missed Feedback, & Missing Feedback*

Feedback is intended to inform users of the results of their actions. This was a common source of confusion for participants; they *misinterpreted feedback* in ways that agreed with their incorrect mental models of the games. Sometimes participants *missed feedback* because they were looking for it in the wrong places. Other times, participants' button presses produced no effects and the games were *missing feedback* that might have helped them to understand what happened.

**Misinterpreted Feedback:** Participants misinterpreted feedback when they misunderstood the meaning of the games' outputs. Automated avatar actions sometimes produced outputs that participants misinterpreted as validating their erroneous mental models. P3 suspected that the 'control' key might make her *Ninja Showdown* avatar use the Sword, pressed it, became surprised that no Sword appeared, and said "*I'm thinking Bomb, where is the Sword?*" In the next round, still confused about why she did not see the Sword, P3 looked at her opponent, noticed that it was using the Sword, and said "*Yeah, Sword is now there.*" P3 misinterpreted her opponent's attack as feedback indicating that she was actually in control of her opponent and successfully commanded him to use the Sword.

**Missed Feedback:** Participants missed feedback when they attended to the wrong parts of the games' outputs. P2 knew that she could make her avatar attack in *Spelunky*, but did not know about the other actions she could control. She pressed the spacebar to make her avatar throw a rope and a bomb during the

tutorial, but when asked why she pressed the spacebar, P2 said “*The space is for attack.*” P2 could not make sense of the Option cues, informing her of which action her avatar would do, and so missed this feedback.

**Missing Feedback:** The games were missing feedback when they produced no feedback informing participants of the effects of their actions, or the lack thereof. Much of P3’s confusion about *Ninja Showdown* seems to have been caused by missing feedback in the tutorial. P3 pressed the spacebar while her avatar was already using the Bomb, which indicated to her that she could make her avatar use the Bomb. There was a critical lack of feedback from the game that might have informed P3 that her spacebar press had no effect.

#### 7.2.5.2 Wrong Expectations: *Inherited Expectations, Incorrect Mappings, & Wishful Thinking*

Participants had the wrong expectations when they expected the games to work in ways that they did not. We found that some participants inherited expectations from other systems they understood, as they expected the games to work in similar ways. Some participants came up with *incorrect mappings* inspired by arbitrary associations between the actions they wanted to control and the buttons they believed should control these actions. Perhaps the greatest source of confusion for participants was their desire to make their avatar do what they wanted it to do and their *wishful thinking* when interpreting the games’ outputs in ways that satisfied this desire.

**Inherited Expectations:** Participants inherited expectations when they expected the games to work like other technologies they already understood. Both

P1 and P5 immediately shadowed the *Spelunky* avatar's jumping out of "instinct" (P5) and may have inherited an expectation that they could make their avatar jump from platformers they played as children. For other participants, who were unfamiliar with platformers, their expectations for how the games worked seem to have been drawn from their expectations of other technologies. P8 was shocked to discover that her *Ninja Showdown* avatar was not as 'smart' as she expected and would sometimes choose the losing weapon. When P8 tried to intervene and make her avatar choose a different weapon, she pressed the 'enter' key because "***Enter does everything.***" Each participant approached the games with their own set of expectations informed by the interactive systems they already knew.

**Incorrect Mappings:** Participants came up with incorrect mappings when they hypothesized that buttons controlled actions because of an arbitrary association between them. P2 wanted her *Spelunky* avatar to jump over and avoid snakes, so she repeatedly tapped the spacebar and asked "***I want to jump... Is there any way to jump?***" Shortly after, P2 said "***I tried to press 'j'***" and when asked why she chose the 'j' key she said "***The game didn't tell me that but I just wish... I just think 'j' would be for jump.***" Just as she had in *Ninja Showdown*, when she hypothesized that she could make her avatar use the Bomb and the Dart with the 'b' and 'd' keys, P2 invented an incorrect mapping between the first letter in the name of the action she wanted her avatar to do and the key for that letter on the keyboard.

**Wishful Thinking:** Participants were given to wishful thinking when they believed that they could control actions that they could not because they wanted to

control these actions. While describing what about the avatar's behaviour was frustrating, P8 said "*Unless I have opportunities I don't know about... If she's not [trustworthy] then I'm like 'uh, am I missing something in the game? Is there another key that I have access to that I don't know about?*" Being unable to make their avatars perform specific actions made participants frustrated and prompted them to think wishfully about how the games were controlled. P8 went on to explain how her desire to make the her avatar use the Bomb led her to believe that she could. She said "***I thought I could, but I couldn't, when she needed to, but wasn't. 'Cause I figured, if the goal is to win and she can't be trusted to do the right thing, there's gotta be somebody who can do it and I'm the only other person here.***" P8's desire to make her avatar do specific actions led her to expect that she could.

All ten participants were able to successfully play the games. However, we found that partial automation made participants confused about how to control the games. Participants' frustration with being unable to fully control their avatars led them to spend a significant amount of their playtime trying to control actions that they could not. They came up with convoluted ways of playing that, because they did not actually produce the effects participants expected, would only appear to work for so long. Participants had to continually revise their mental models of the games and consequently reinterpret the same outputs. Our grounded theory of automation confusion describes the types of confusion participants experienced. It explains how participants' expectations made them confused by the games, how their confusion made them frustrated, how their frustration made them behave differently, and how their behaviours led them to misunderstand the games in new ways. These

observations yielded several insights into the design of partially automated games for non-gamers, which we explain in the next section.

## 7.3 Design Implications for Partial Automation

We designed both games with guidance from the literature, including tutorials and awareness cues to help players understand the copilot, but our results suggest that more guidance is needed. Our analysis identified several issues in the design of the games (e.g., unclear feedback & insufficient training) and the copilot (e.g., ceding control & cooperation) that we believe can be addressed. In this section, we provide recommendations for the design of partial automation for non-gamers based on our observations.

### 7.3.0.1 Increase Control

Partial automation should afford players as much control over their avatars as possible. The partially automated games used in our study were designed to help non-gamers' play successfully by reducing their control. However, this confused participants if they wished to make their avatars perform actions which were not under their control. Instead of removing control altogether, partial automation could allow both the player and the copilot to control all of an avatar's actions at the same time, falling back on the copilot for support when the player fails to act. This approach may scaffold non-gamers' learning by allowing them to choose which mechanics they control. This form of support is analogous to training wheels [45]

or a stencil [154], where the support is provided only when necessary, and could be removed altogether if the player becomes proficient in the game.

### 7.3.0.2 Beware Misinterpretation of Feedback

Designers of partially automated games should rigorously test the comprehensibility of their games' feedback with target players. *Ninja Showdown* and *Spelunky* provided awareness cues designed to improve participants' understanding, but participants often misinterpreted the meaning of these cues. Confused participants continually revised their mental models of the games and so ascribed different meanings to the same outputs over the course of play. Had the meanings of these cues been reinforced some other way, for example having Emi say "*I think I'll use a Dart this time*" (instead of just showing a Dart icon), then participants might have more faithfully interpreted what the cues were designed to convey.

### 7.3.0.3 Provide Training

Partially automated games should provide training for non-gamers to front-load their learning and thereby minimize the learning they do during play. Instructing players on how to play has fallen out of vogue for game design, in favor of gameplay tutorials that teach players through play. However, we believe that training participants on what they could control might have prevented some of their confusion. Automated games could show players what to do in a variety of gameplay situations and also quiz players about what is happening in the game to verify their

understanding. Training was specifically requested by both P3 and P8, who suggested that video game courses be made to teach them how to play the games their children play.

#### 7.3.0.4 Tell Players How to Cooperate

Partially automated games should teach players how to cooperate with their avatars by telling them what to do when they dislike their avatar's behaviour. In *Ninja Showdown*, most participants understood what to do when their success was determined entirely by them. They often knew, for example, that they should always make Emi use the Sword when Takeshi used the Bomb, because Swords beat Bombs. However, some participants were less sure what to do when their success was dependent on the copilot. When Takeshi used the Sword, pressing the spacebar would force a tie, while not pressing the spacebar would result in either a win or a loss. Most participants wanted to win. P6 wanted additional actions she could do to "save Emi" and P9 was frustrated that there was no way to "defend yourself". Participants had significant difficult learning how to cooperate with the copilot and may have benefited from being told what to do when they needed the copilot to act.

Our analysis of non-gamers' automation confusion indicates that partial automation can improve a game's accessibility to non-gamers, but also that partial automation must be rigorously tested with target players. Awareness cues can help non-gamers to recognize and predict what the copilot does, but the way that awareness information is presented may be more comprehensible by some players than by others. If the meanings of these cues are not obvious at a glance or they do not

display the information that players need, then players can get caught in a frustrating metagame of trying to figure out how the game is controlled through trial and error. The design guidelines proposed in this section may help to dispel or prevent players' misunderstandings, but properly addressing players' automation confusion may require multiple costly rounds of iterative redesign.

## 7.4 Summary

We have demonstrated how automation confusion can occur when non-gamers play partially automated games. In Chapter 2 we presented the game interaction model (i.e., *receive stimuli*, *determine response*, and *provide input*) and in Chapter 6 we explained, in reference to this model, why one participant superstitiously tilted a bite switch to steer his avatar. We suggested that input provided by partial automation circumvented the causal link between the inputs players provide and the stimuli they receive, which may lead players to misunderstand why some in-game events occurred and develop bad mental models. The results presented in this chapter lend evidence to this interpretation but they also indicate that there is more going on.

We saw repeatedly how players' attitudes towards the games affected the criticality with which they interpreted stimuli and also how players' desire to perform specific actions led them to misinterpret stimuli in favorable but incorrect ways. Our initial account of automation confusion was overly concerned with the rational aspects of mental model development and did not consider the critical role that emotion and attention play. It appears that illusions of control, curious rituals, confirmation biases, and the like may be the products of players' necessity in the moment, rather than pure reason. Therefore, the grounded theory proposed in this

chapter provides a more comprehensive account of automation confusion that may help designers to identify and address automation confusion in players. However, it does not provide a general solution for the problem of automation confusion and, since automation confusion originates partly in players, it is unlikely that any such solution can exist. Ultimately, automation confusion is a design problem that must be addressed iteratively through user evaluations and thoughtful refinement. This is a potentially significant hindrance to the use of partial automation in games.

This does not mean that improving games accessibility with partial automation is a lost cause; partial automation in our study games really did make these games more accessible to participants. Instead, a major takeaway from the results of our automation confusion study is that partial automation must be engineered in a way that is robust to the radical changes in its design. It may be that a standard partial automation architecture can help developers to separate partial automation's functionality into independent units and encapsulate its functionality in ways that promote reuse. In this way, changes to the design of partial automation may be less costly, as they would have less of an impact on its implementation. In the next chapter, we will propose a novel reference architecture and an associated toolkit to facilitate the development of partial automation.

## **Part III**

### **Discussion & Conclusion**

# CHAPTER 8

---

## Partial Automation Architecture & Toolkit

---

WE have shown how partial automation can improve the accessibility of digital games. In Chapter 2, we identified two groups of players who face disabling barriers to access in games: players with motor impairments, who may be unable to provide input using a game's default controllers, and non-gamers, who may be unable to determine responses and provide input quickly enough to keep up with fast-paced games. As explained in Chapter 3, there are several approaches designers can use to make games more accessible, but ultimately the efficacy of each approach has limits. In Chapter 6, we demonstrated how personalizable partial automation can extend the accessibility afforded by these other approaches and enable players with motor impairments to play games they would be unable to play otherwise. As explained in Chapter 4,

interactive systems providing human-AI shared control, such as partial automation, have been designed for use in at least six application domains and shared control systems must be designed with consideration for several well-known problems of human-machine cooperation. In Chapter 7, we showed how partial automation can help non-gamers to play fast-paced action games, but we also found that partial automation for non-gamers can cause automation confusion if not designed iteratively with target players. Despite this potential confound, partial automation remains a promising approach to broadening a game’s accessibility that can be applied in any game.

Due to partial automation’s novelty and limited use in only a handful of games, there exist no tools to help developers implement partial automation. The design guidelines proposed in Chapter 7 suggest that issues such as automation confusion can be overcome by iteratively redesigning partial automation with target players; however, these guidelines do not consider the impact that changing a partial automation design might have on development. There are many ways that developers could realize partial automation and not all of them allow rapid iteration.

In developing large commercial games, modifying how partial automation is implemented might be too costly to justify. Interviews with game developers discussing games accessibility have revealed that “*the prospect of having to reengineer solutions to the same set of problems over and over is daunting and unattractive.*” [246] Game developers largely do not understand accessibility and are unwilling to implement solutions to all but the most simple of accessibility issues (e.g., subtitles for deaf players & color palettes for color-blind players). As one developer put it: “*With things like motor impairment, when one person’s needs can be so different*

*from the next, and the same solution won't work for both, it's a lot harder.”* Having implemented partial automation in over ten games, including many prototypes that were not used in our studies, we have identified several architectural features that can facilitate the development of partial automation. We believe that a reference architecture and an associated development toolkit will make implementing partial automation more tractable for developers and reduce the impact of iterative design on development.

This chapter is about our novel Concordia architecture and development toolkit. Much like the shared control design patterns presented in Chapter 5, we have identified software design patterns in the partial automation we developed and devised a reference software architecture that supports these patterns. The purpose of the Concordia architecture is:

1. to reduce the overhead of designing and implementing partial automation
2. to allow the reuse of code implementing common partial automation functionality
3. to support iterative refinement by encapsulating partial automation functionality that is likely to change during development

In the next section, we will introduce the Concordia architecture and explain how it structures partial automation functionality.

## 8.1 The Concordia Architecture

The Concordia architecture provides a template structure for partial automation that is applicable in any game. It separates partial automation functionality into

five types of plug-ins that can be easily customized for use in one specific game or across multiple games using the same game engine (e.g., Unity, Unreal, or Godot). Primarily, using the Concordia architecture serves to support the modification and reuse of partial automation software, while using the associated toolkit provides additional functionality that developers need not implement. In particular, games using the Concordia toolkit support control remapping and allow players to simultaneously use multiple controllers without requiring developers to write any additional code. Since the presentation of awareness cues is a core aspect of the Concordia toolkit's functionality, games using the toolkit can also automatically determine which awareness cues should be presented. In the rest of this section, we will explain why we made these design choices and more precisely specify the Concordia architecture.

### 8.1.1 Requirements

Our findings from the studies presented in Chapters 6 & 7, as well as our experiences of implementing partial automation in over ten games, have helped us to identify a set of requirements for a partial automation reference architecture. We believe that a reference architecture should support the *personalizability*, *interpretability*, *expressivity*, and *modifiability* of partial automation in games. We define these terms below.

**Personalizability:** The results of the personalizable partial automation study (Chapter 6) indicate that players highly value customizing the set of devices they use to play games. Participants believed that allowing players to use alternative devices, use only the accessible parts of devices, use multiple devices

simultaneously, and delegate control of inaccessible inputs to partial automation would greatly improve motor-disabled players' access to games. The two participants who played using the joystick and bite switch would have been unable to play without this level of personalization and they said that using familiar devices helped them to play. We, therefore, consider personalizability to be a requirement for partial automation. Players must be allowed to choose which inputs they control, what devices they use, and which parts of these devices are used to control the game. Therefore, the personalizability of games created using a partial automation reference architecture should be supported.

**Interpretability:** As explained in Chapter 4, users of automated systems need to maintain their awareness of the copilot to cooperate with it effectively. The results of the automation confusion study (Chapter 7) indicate that awareness cues can help players to make sense of what the copilot is doing and potentially improve their mental models through critical observation. While there is some chance that poorly designed awareness cues will lead players to misunderstand how the game works (e.g., *misinterpreted feedback*), it is widely recognized that the decision-making of automated systems must be interpretable by users. For this reason, we consider it a requirement that partial automation be interpretable by players through the presentation of awareness cues exposing the copilot's state. Therefore, the interpretability of the copilot's state in games created using a partial automation reference architecture should be supported.

**Expressivity:** Partial automation is a generally applicable approach to games accessibility with almost no commitments related to how it is implemented. The copilot must be able to provide input to the game, but any AI algorithm can be used in games to determine what input the copilot provides. Input from the player and the copilot also needs to be merged, but multiple mediation strategies have been used for different forms of assistance (e.g., one-switch automation and player balancing). Developers implementing partial automation need to be able to customize its functionality to express a wide variety of designs, including designs demonstrated in existing games using partial automation. Therefore, the expressivity of partial automation in games created using a partial automation toolkit should be supported.

**Modifiability:** Since the copilot considers the game’s current state when making decisions, developers may choose to implement partial automation at the level of the game’s other source code. They may integrate some of the partial automation functionality into existing modules or allow the copilot to access and modify data that is shared with other parts of the game’s code. These sorts of design choices might seem cohesive or efficient at the outset, however there is a risk that later modifications incited by player feedback might needlessly impact functionality unrelated to partial automation. To avoid this, partial automation functionality must be encapsulated and separated from the game’s other source code in a way that supports modification. Designers must be able to rapidly iterate on their partial automation designs and developers must be able to implement revisions without adversely affecting other game

functionality. Therefore, the modifiability of partial automation created using a reference architecture should be supported.

These requirements have informed the design of the Concordia architecture and development toolkit. By adhering to the Concordia architecture, developers can create easily modifiable partial automation that is personalizable and interpretable by players. Using the Concordia development toolkit ensures that partial automation functionality is separated from other game functionality and reduces the need for developers to implement boilerplate personalization and awareness conferring functionality themselves. In the next section, we present the Concordia architecture and describe how it is realized in the Concordia toolkit.

### 8.1.2 Architecture

Concordia is a reference architecture for partial automation that was inspired by the most successful design choices made while creating the games used in our studies. Designing partial automation involves deciding which inputs should be automated, what degree of personalization should be supported, what strategy the copilot should follow, how the copilot should behave to enact its strategy, what awareness information should be presented to players, and how this information should be represented as awareness cues. Within the Concordia architecture, functionality related to each of these design choices is encapsulated by different plugins. The major benefit of this approach is that functionality that can be separated is implemented separately, so that modifying part of the partial automation does not needlessly impact other parts. If players are confused by a game’s awareness cues, then perhaps only the awareness functionality needs to change. If players

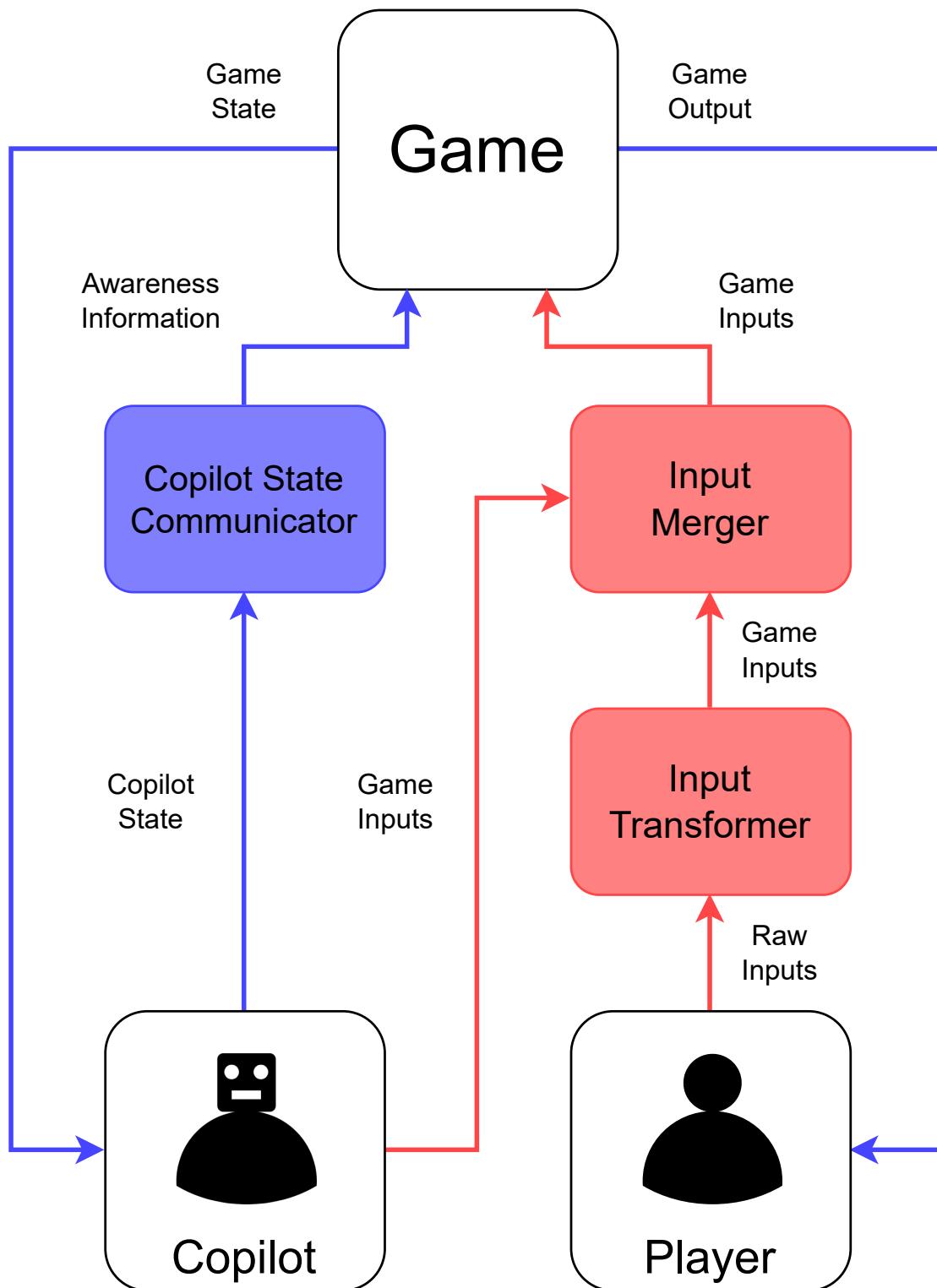


Figure 8.1: The Concordia abstract architecture. Arrows represent data flow between modules. Data flow related to providing input to the game is red, while data flow related to the game's state are blue.

dislike the copilot's behaviour, then the way that it provides input to the game can be changed without affecting its decision-making.

The core insight of the Concordia architecture is that the copilot's decision-making functionality should be separated from functionality related to providing input to the game and presenting awareness information to the player. The copilot's decision-making data represent what it is currently doing and intends to do in future, so these are the data that players need to maintain awareness of the copilot's activities. Concordia presents information about the game's state to the copilot directly and to the player in the form of stimuli (e.g., audio and video). Information about the copilot's state is included in these stimuli. Input from the player's input devices is gathered and merged with input provided by the copilot to control the game.

The high level architectural diagram in Figure 8.1 shows the flow of data from the player and an assistive copilot AI to the game and then back to the player and the copilot. Red paths depict how the player and the copilot provide input to the game. An *Input Transformer* is needed to gather input from the player's devices and transform it into game inputs. For example, a generally applicable Input Transformer might transform input from an eye tracking system, a power wheelchair head controller, or a similarly unconventional gaming device into analog stick input for controlling avatar movement. Additionally, an *Input Merger* is needed to merge game input provided by the player and the copilot. An Input Merger for buttons, for example, might provide input representing a button press when either the player or the copilot provide that same input. Blue paths depict how the game's state is communicated to the player and the copilot. Using game state

information supplied by the game, the copilot decides what to do and a *Copilot State Communicator* is needed to present the copilot's decisions to the player in the form of awareness cues. Since awareness cues are part of the game's output, players receive stimuli informing them of what is happening in the game and in the copilot.

As a plug-in architecture, Concordia is applicable in any game and is capable of expressing all of the partial automation designs we have created and encountered. It provides a structure for partial automation that may facilitate the reuse of common functionality across multiple games. Integrating Concordia into a game requires only minimal modification to the game's existing code; the game needs to expose its state to the copilot, it needs to present awareness information to the player as stimuli, and it needs to receive merged game input. Adherence to the Concordia architecture is enforced by its plug-in system, which allows developers to customize the partial automation's behaviour by registering game-specific plug-ins.

### 8.1.2.1 Types of Plug-ins

The Concordia architecture (Figure 8.2) organizes partial automation functionality into five components: *Copilot AI*, *Automation*, *Awareness*, *Control*, and *Mediation*. Each component is responsible for a different aspect of the partial automation and each contains a collection of plug-ins that implement component-specific functionality, as described below.

**Processors:** With consideration for the game's current state, *Processors* make the copilot's gameplay decisions. This would include, for example, deciding which

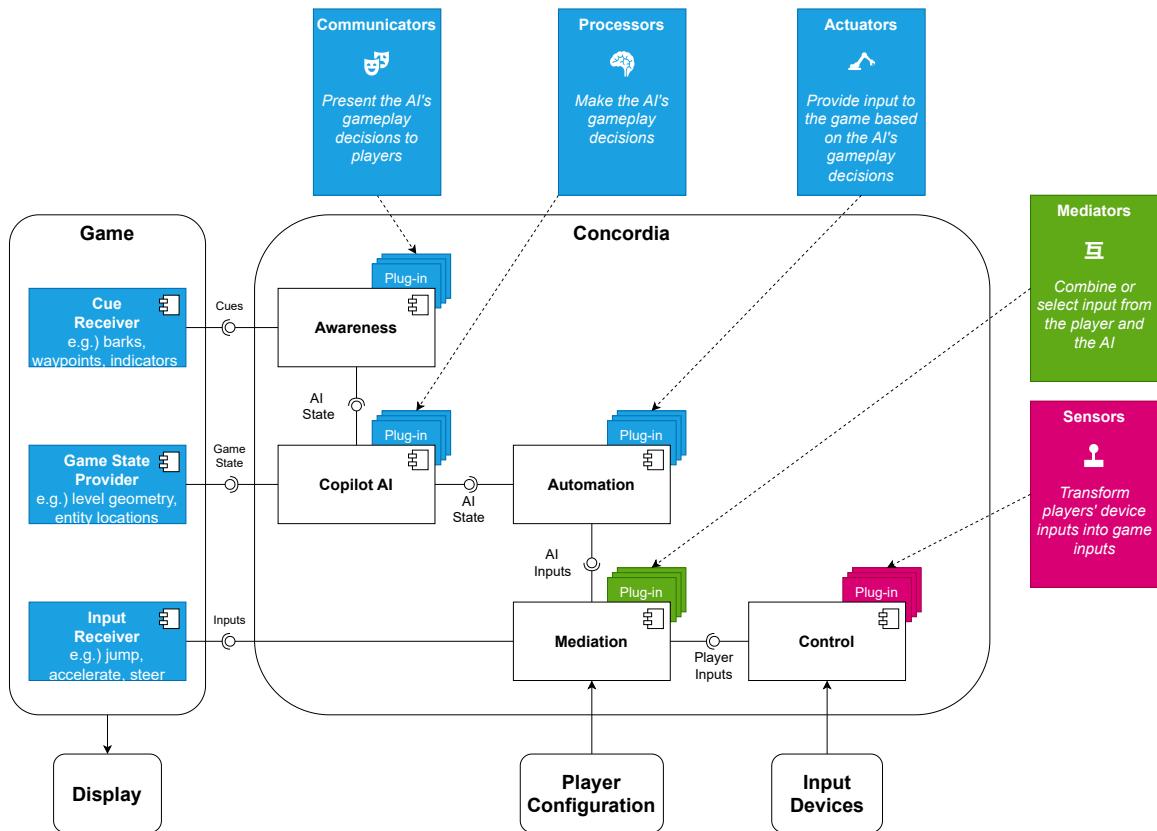


Figure 8.2: The Concordia reference architecture. Concordia requires an interface exposing the game's state and provides interfaces exposing awareness information to be presented as awareness cues as well as merged game inputs from the player and the copilot. Each of the Concordia architecture's components, as well as the interfaces they provide and require, are shown. The plug-ins that each component uses are coloured according to the situations in which they need to be modified. **Blue plug-ins** change in response to changes in the game. **Green plug-ins** change in response to changes in the types of inputs. **Red plug-ins** change in response to changes in the types of inputs or the runtime environment.

enemy to attack and which attack to do to it. The benefit of encapsulating the copilot’s decision-making functionality is that it separates the copilot’s strategic functionality—deciding what to do—from its tactical functionality—deciding how to do it. Processors can be registered with the Copilot AI component.

**Communicators:** Using the data generated by the Copilot AI component, *Communicators* present the copilot’s gameplay decisions to players. For example, a Communicator might take the path generated by a pathfinding Processor and produce a vector representing the copilot’s intended movement direction. The benefit of encapsulating awareness functionality is that it allows Concordia to automatically determine which awareness cues need to be displayed. Communicators can be registered with the Awareness component.

**Actuators:** Also using the Copilot AI component data, *Actuators* provide input to the game based on the copilot’s gameplay decisions. For each of the game’s inputs, there must be an Actuator that automates control of that input; for example, a button that makes the player’s avatar jump would need an Actuator that produces a boolean value representing a button press. The benefit of automating control of each game input using a separate Actuator is that it affords personalization by allowing players to assign control of any input to partial automation. Actuators can be registered with the Automation component.

**Sensors:** By gathering human interface device data from the game’s runtime environment (e.g., operating system, web browser, etc.), *Sensors* transform players’ device inputs into game inputs. A Sensor might, for example, transform accelerometer data into analog stick data used to control an avatar’s movement. As we will explain later in this section, the benefit of encapsulating player input functionality is that it allows Sensors to be reused across games and engines. Sensors can be registered with the Control component.

**Mediators:** Using the inputs provided by the Control and Automation components, *Mediators* combine or select input from the player and the copilot. As explained in Chapters 3 & 4, partial automation comes in two forms (i.e., input automation & one-switch automation) and shared control systems can combine or select commands. By creating Mediators, developers can implement these different forms of partial automation and mediation strategies; for example, a Mediator might provide aim assistance by blending aiming commands or prevent the player from performing actions that the copilot does not also want to perform. The benefit of encapsulating mediation functionality is that it allows the reuse of generally applicable mediation strategies across games. Mediators can be registered with the Mediation component.

These plug-ins are created by developers to implement game-specific or engine-specific functionality in a way that is easily customizable, modifiable, and in some cases reusable. As shown in Figure 8.2, each of the Concordia architecture’s components provide an interface to other specific components. The Copilot AI component provides an *AI State* interface to the Automation and Awareness components. The

Automation component provides an *AI Inputs* interface to the Mediation component and the Control component provides a *Player Inputs* interface to it as well.

### 8.1.2.2 Plug-in Reuse and Modification

As a plug-in architecture, Concordia affords a high degree of customizability and is therefore applicable in any game. The fixed structure of the Concordia architecture's components, shown in Figure 8.2, encapsulates disparate partial automation functionality in a way that allows developers to create easily modifiable plug-ins that may be reused across games and engines. This may be helpful during design and development because Concordia plug-ins may need to be modified in response to changes in the game, the types of inputs, or the runtime environment.

If, for example, a new mechanic were introduced into a game during development, then the copilot's decision-making functionality may need to change and a Processor may need to be created or modified. Processors, Communicators, and Actuators all implement game-specific functionality; while some of these plug-ins may be reusable across games with mechanical similarities, they may need to be modified in response to changes in a game's design.

In contrast, Mediators operate exclusively on input to the game and may therefore be applicable across games and engines that share the same types of inputs (e.g., buttons, axes, etc.). Developers may choose to create their own Mediators, but doing so is only necessary when a game has inputs that no existing Mediator can handle. If, for example, touchscreen input were added to a game, then new Mediators for tapping and swiping may need to be created.

While most Concordia plug-ins only need to be modified in response to changes to a game’s design, Sensors may need to be created or modified due to changes in the game’s runtime environment. Since gathering input from the player’s devices may involve calls to an operating system interface (e.g., Win32 API), a game may need different Sensors for each operating system. If, for example, a console game were ported to iOS, then a new Sensor that interprets touchscreen input as gamepad input may need to be created.

## 8.2 The Concordia Toolkit

To demonstrate the utility of the Concordia architecture, we have implemented an associated toolkit<sup>1</sup> in C# and used it to create partial automation in a *Space Invaders* clone called *Interface Invaders*<sup>2</sup>. The Concordia toolkit is a library that provides extensible classes for all of the Concordia architecture’s components and different types of plug-ins. Since the toolkit is written in C#, it is compatible with game engines that use the .NET Framework, such as the popular Unity game engine used to create *Interface Invaders*. As explained in Section 8.1.2.2, some Concordia Plugins may be reusable across games, so the Concordia toolkit provides example Mediators and Sensors compatible with Unity’s legacy input system. In this section, we will describe how we implemented partial automation in *Interface Invaders* using the Concordia toolkit and evaluate the toolkit in terms of its expressiveness and the potential source code impact of modifications.

---

<sup>1</sup>The Concordia toolkit is publicly available on [GitHub](#)

<sup>2</sup>*Interface Invaders* can be played in a web browser using this [link](#)

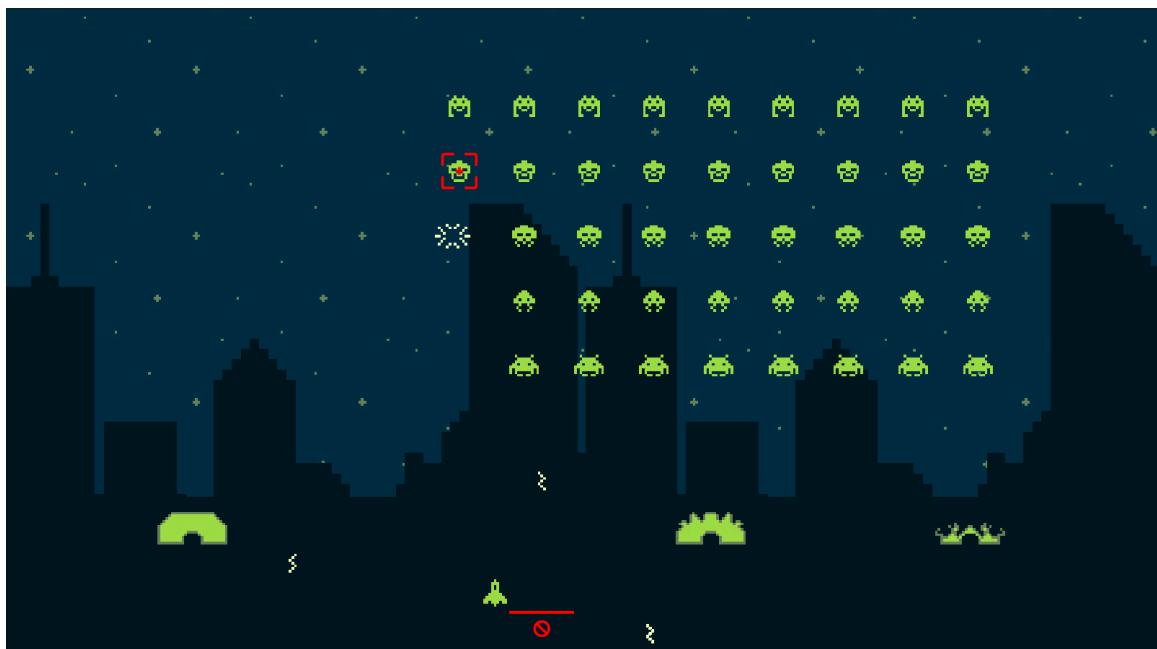


Figure 8.3: *Interface Invaders*. The space ship at the bottom of the screen is the player's avatar and the shapes above are the enemies. The red box around the enemy on the left is an awareness cue indicating which enemy the copilot has chosen to target. The red line below the player's avatar is an awareness cue indicating that the copilot is avoiding the laser above.

### 8.2.1 The Concordia Toolkit in *Interface Invaders*

We created *Interface Invaders* to demonstrate use of the Concordia toolkit in a well-known and simple game implemented with Unity (Figure 8.3). Just as in *Space Invaders*, players control an avatar at the bottom of the screen that can move horizontally and shoot lasers towards the top of the screen. The game has two inputs: *Shoot*, which is a button input (i.e.,  $\text{Shoot} \in \{\text{True}, \text{False}\}$ ) that causes the player's avatar to shoot, and *Move*, which is an axis input (i.e.,  $\text{Move} \in [-1, 1]$ ) that moves the player's avatar to the left and right. A swarm of enemies move in unison across the screen and then down towards the bottom, shooting lasers back at the player. Barriers above the player's avatar can block a finite amount of lasers before they are destroyed, while it takes only one laser to destroy an enemy or the player's avatar. The player loses the game if they are hit by a laser or the swarm of enemies reaches the bottom of the screen. The player wins the game if they destroy all of the enemies.

From this description, it should be obvious that several Concordia plug-ins will be needed to create partial automation in *Interface Invaders*. The game has two game inputs, called *Shoot* and *Move*, and therefore requires an Actuator to provide AI input for each game input. These game inputs have different types, so Sensors for gathering these types of inputs from the player's input devices and Mediators for merging AI and player input will be needed. Since button and axis inputs are supported by Unity's legacy input system, we used the button and axis Sensors and Mediators provided with the Concordia toolkit. With those in place, we created game-specific Processors, Actuators, and Communicators.

### 8.2.1.1 Processors and Actuators in *Interface Invaders*

Overall, the partial automation's strategy is to identify which enemies can be hit, select a target from among those, and avoid positions where the avatar will be hit by a laser. We, therefore, created three Processors: one for identifying where the avatar can be hit, one for identifying where enemies can be hit, and one for selecting which enemy to target. Collectively, these Processors reference four aspects of the game's state: the location of the *avatar*, the locations and movements of the *enemies*, the locations of the remaining *barriers*, and the locations of the enemies' *lasers*. We then created two Actuators, one for each game input. The *Shoot* Actuator provides input representing a button press any time the avatar is in position to hit its chosen target or the avatar is far from its target and able to hit another enemy. The *Move* Actuator provides input that causes the avatar to move towards the location where it can hit its target so long as doing so does not cause it to move into the path of an enemy laser. These plug-ins enable the partial automation to play effectively enough to win, but its behaviour may surprise or frustrate players without some explanation.

### 8.2.1.2 Communicators in *Interface Invaders*

Players may not understand why the partial automation sometimes chooses not to shoot when it can hit an enemy or why the partial automation stops to wait for some lasers to pass, but not others. To support players awareness and understanding of the partial automation's behaviour, we created two awareness cues designed to represent its intention to hit its target and avoid nearby lasers (Figure 8.3). The target Communicator produces the location of the target and its associated awareness cue displays a red box around that enemy. The laser Communicator produces

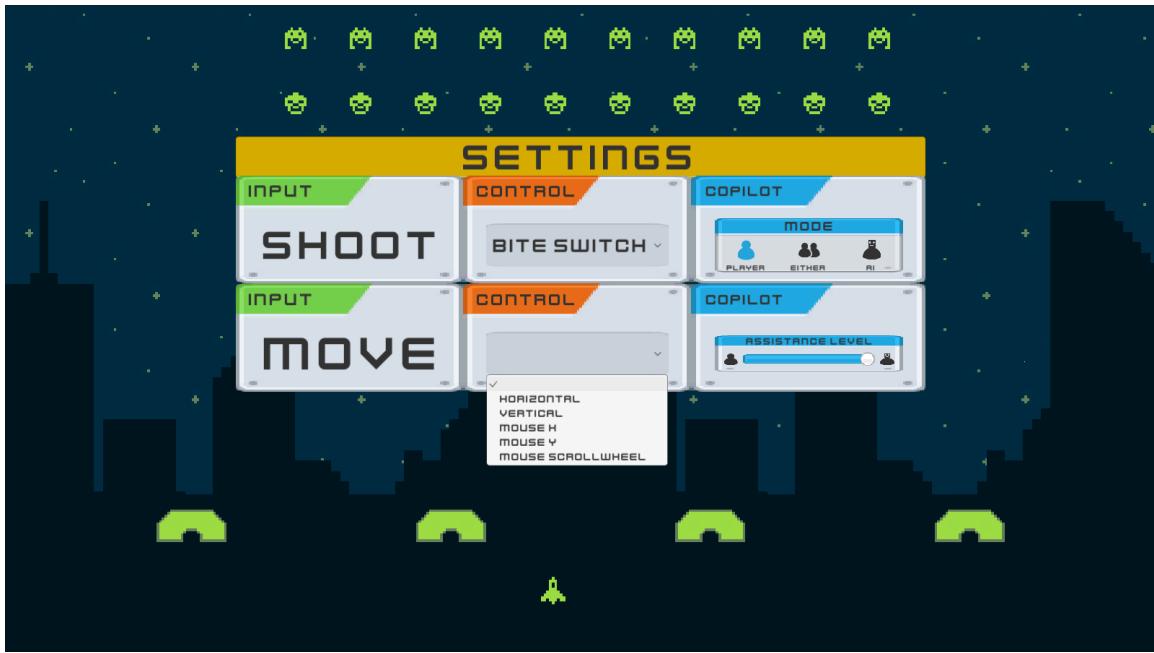


Figure 8.4: The Concordia settings menu for *Interface Invaders*. In this example, the player controls the Shoot input using a bite switch while the copilot controls the Move input.

a collection of locations near the avatar where it would be hit and the associated awareness cue displays a red icon at those locations. These awareness cues are only presented if the AI state data they represent were referenced by Actuators used to control the game. Because the target Communicator references the target, as do the *Shoot* and *Move* Actuators, the target awareness cue is displayed when partial automation controls either input. In contrast, the laser Communicator references the locations of the lasers, which is referenced by the *Move* Actuator but not the *Shoot* Actuator, so the laser awareness cues are only displayed when the player is assisted with the *Move* input.

### 8.2.1.3 Sensors and Mediators in *Interface Invaders*

Players can personalize the partial automation in *Interface Invaders* using the settings menu. As shown in Figure 8.4, the settings menu displays the names of each game input along with a dropdown menu used to assign a player input to control it and a panel used to configure its Mediator parameters. Since the game's two inputs have types that are supported by Unity's legacy input system (i.e., buttons and axes), our partial automation in *Interface Invaders* uses the Sensors and Mediators provided with the Concordia toolkit.

The provided button Sensor produces Plugins for all of the gamepad, mouse, and keyboard buttons available to the legacy input system as well as Plugins that interpret axis inputs as button inputs; this allows the player to shoot by pressing the spacebar, biting a bite switch, or tilting an analog stick. The provided axis Sensor produces Plugins for all of the gamepad and mouse axes available to the legacy input system; this allows the player to move their avatar by tilting an analog stick, moving their cursor, or rotating a scroll wheel. Mediators for these types of inputs are provided with the Concordia toolkit as well and each exposes a set of parameters for player configuration.

The provided button Mediator has three operational modes: *Player*, which returns input exclusively from the player, *AI*, which returns input exclusively from the copilot, and *Either*, which returns input representing a button press (i.e., True) when either the player *or* the copilot provide that same input and False otherwise. The provided axis Mediator has one real-valued parameter that is bounded within 0 to 1 and used to blend axis input from the player and the copilot. Dragging the slider to the right increases the copilot's contribution, up to the point that the copilot

controls the axis exclusively, so this parameter represents the degree of assistance provided by partial automation for this input.

Using these plug-ins, *Interface Invaders* provides a form of partial automation that is designed to be personalizable and interpretable by players. The game allows players to choose which parts of their input devices they use to play and allows them to configure the assistance they receive from partial automation. As we saw in Chapter 6, a high degree of personalizability can broaden the accessibility of partially automated games to players who can control some of the game’s inputs using alternative devices. The awareness cues in *Interface Invaders* communicate information players need to make sense of the partial automation’s behaviour and these cues were designed to be easily interpretable by players. However, as we saw in Chapter 7, players may misinterpret awareness cues and ascribe unintended meanings to them, so the cues and Communicators in *Interface Invaders* may require modification. In the next section, we evaluate the Concordia toolkit’s capacity to support these sorts of modifications and express a variety of partial automation designs.

### 8.2.2 Concordia Toolkit Evaluation

Concordia and its associated toolkit provide a template structure for partial automation that is applicable in any game, as it allows developers to create plug-ins that implement game-specific functionality. There are no restrictions on the computations performed by these plug-ins, only the data that they reference, since the toolkit

can evaluate any computable function. This allows developers to express a wide variety of partial automation functionality in a way that upholds our requirements that partial automation be personalizable, modifiable, and interpretable. These are our requirements for the Concordia architecture (Section 8.1.1) and in this section we determine whether these requirements are met by the Concordia toolkit.

### 8.2.2.1 Personalizability

When provided with a suitable set of Sensors and Mediators, players can personalize how they control games that use the Concordia toolkit. The default input assignment functionality provided by the Mediation component enables players to choose which compatible input devices they use and what parts of these devices they use to control specific game inputs. By handling input assignment in a generic way, Concordia affords personalization without the need for developers to write boilerplate input remapping functionality.

### 8.2.2.2 Interpretability

Concordia encapsulates awareness conferring functionality within the Awareness component, which gives Communicators access to information about the copilot's state. Since awareness cues are designed to inform users of their collaborators' actual and intended effects on their shared workspace, these are exactly the data Communicators need to communicate. By writing game-specific Communicators, games using the Concordia toolkit can integrate any form of awareness cue into the game's output.

### 8.2.2.3 Expressivity

We consider the Concordia architecture to support expressivity if it supports the creation of a wide range of the behaviours described in previous work [259]. For the Concordia toolkit to be considered sufficiently expressive, it must be capable of implementing the forms of partial automation found in existing partially automated games.

**Input & One-switch Automation:** The Concordia architecture's Mediation component uses separate plug-ins for each game input, which allows developers to implement any computable form of input automation. It is also possible to implement partial automation that adheres to the one-switch automation design pattern, but doing so requires the creation of special Mediators and some configuration by the player. Implementing the one-switch automation in *Spelunky*, for example, might involve creating a button Mediator that provides input representing a button press whenever that same input is provided by *both* the player and the copilot. By assigning the same button to control multiple game actions (e.g., jump, attack, rope, bomb, etc.), the player can press the button to cause their avatar to perform whichever action the copilot intends to perform.

**Player Balancing:** As explained in Chapter 3, player balancing helps weaker players to compete with stronger players and some forms of player balancing can be implemented as input automation. In a shooting game, for example, partial automation could provide a player-configurable degree of assistance with

the aiming input or it could replace the player’s aiming command to aim directly at the closest opponent when the player presses the fire button. These forms of player balancing can be implemented using the Concordia toolkit by creating specialized Mediators.

**AI Algorithms:** As explained in Chapter 3, any AI algorithm could be used to implement partial automation’s decision-making and automation functionality. This holds true of partial automation implemented using the Concordia toolkit, as it is possible to create Processors and Actuators that compute any computable function. A developer could, for example, create a Processor that classifies gameplay situations using a neural network and an behaviour tree Actuator that considers these classifications when deciding what to do.

#### 8.2.2.4 Source Code Impact

Designing usable partial automation may involve multiple rounds of iteration that necessitate modifications to the partial automation’s source code. We therefore consider it a requirement that partial automation be easily modifiable and limit the impact that design changes have on the partial automation’s source code. *Change impact analysis* [183] quantifies the impact of source code modifications in terms of an estimated impact set (EIS)—the estimated number of elements affected by the change—an actual impact set (AIS)—the number of elements actually affected by the change—and the number of source lines of code (SLOC) that are modified or added during implementation [25; 216]. To evaluate whether partial automation implemented using the Concordia toolkit is easily modifiable and limits the source

code impact of redesign, we changed the gameplay strategy followed by the partial automation in *Interface Invaders*.

Originally, the partial automation in *Interface Invaders* prioritized shooting the enemies in the bottom row of the swarm. Since the player loses when the bottom row reaches the bottom of the screen, these enemies are among the most threatening and may therefore be the enemies players most want to hit. However, more experienced players may find this simplistic strategy frustrating and prefer that the partial automation prioritize the enemies on the side columns over the bottom row. We estimated that this change in strategy would require us to modify only the Processor that selects targets and none of the other plug-ins. In this we were correct; we revised the Processor’s code to target the bottom row only when it is dangerously low and target enemies in the closest side column otherwise. Measures of this modification’s source code impact, generated using David A. Wheeler’s ‘SLOC-Count’<sup>3</sup>, indicate that the modified code’s length (i.e., 90 SLOC) is only 4 lines longer than the original code’s length (i.e., 86 SLOC). This small modification to the partial automation’s code yielded the desired change in its behaviour.

The reason why this modification was localized within only one plug-in and required the addition of so few lines of code is because of the encapsulation of functionality enforced by the Concordia architecture. The partial automation in *Interface Invaders* already contained Actuators and Communicators capable of providing input and presenting awareness information based on the target selected by this Processor. Therefore, changing the way in which the Processor selects targets did not require further modifications to the partial automation’s other functionality. Not all modifications will have this little impact on the partial automation’s source

---

<sup>3</sup>SLOCCount and details of how SLOC is counted are available on [Wheeler’s website](#)

code; for example, adding a new Processor might prompt modifications to several Actuators that need to use it or the addition of another Communicator and an associated awareness cue. But nonetheless, some changes brought on by design may be realized with only minor modifications localized within a small set of related plug-ins.

### 8.3 Summary

We have demonstrated how partial automation can improve the accessibility of digital games and how the development of partial automation can be facilitated by a reference architecture and associated toolkit. Concordia provides a template structure for partial automation that may support the implementation and modification of partial automation that is personalizable and interpretable by players. The Concordia toolkit provides extensible classes that allow developers to create plug-ins that can be customized to express a variety of behaviours and that may be reusable across games and engines. Using the Concordia toolkit may help developers to easily modify existing partial automation code at the pace set by iterative design, supporting the creation of partial automation within a typical game development process. We believe that the Concordia architecture and toolkit will make improving games accessibility more tractable for developers by providing a generally applicable solution.

# CHAPTER 9

---

## Conclusion

---

WE have established that partial automation is a games accessibility technique that can improve the accessibility of digital games. When playing with partial automation, players share control of the game with an assistive AI copilot that controls the game inputs that players cannot control or have difficulty controlling. This can broaden a game's accessibility to players who cannot determine responses fast enough to keep up with fast-paced games and players who are unable to provide input using a game's controllers. The improved accessibility afforded by partial automation may help players to realize the social benefits of playing together and the therapeutic benefits of playing in physical rehabilitation.

For players with motor impairments, who may be unable to use part of a game's controller, partial automation can enable them to play using only the accessible parts of controllers. In Chapter 2, we described how rehabilitation games are designed to improve patients' rehabilitation experiences and motivate them to do exercises that they may consider boring. However, these games are inaccessible to players who cannot perform these exercises and some players with motor impairments may be unable to realize the benefits of rehabilitation gaming. In Chapter 3 we explained that there are several approaches designers can take to extend a game's accessibility further, but also that these approaches have fundamental limitations. If a player is unable to control some part of the game, then they may be unable to play at all unless these parts are removed. In Chapter 6, we showed how personalizable partial automation enabled players with spinal cord injury to play rehabilitation games that they would have been unable to play otherwise. Participants said that partial automation allowed them to use controllers that they found familiar and that playing these games might have improved their most negative rehabilitation experiences.

For non-gamers, who may have difficulty deciding what to do in games, partial automation can simplify how games are played. In Chapter 2, we recounted the reasons why non-gamers may want to play games and also the many barriers non-gamers encounter in games designed for more experienced players. Non-gamers may have difficulty making sense of the stimuli they receive from the game, determining how to respond in different gameplay situations, and providing input using gaming controllers. In Chapter 4, we surveyed six problem domains that use human-AI shared control to improve users' interactions and explained why users'

awareness of automation must be supported in human-machine cooperation. In Chapter 7, we showed that partial automation may help non-gamers to play the sorts of fast-paced action games that are popular among gamers. Participants believed that partial automation would enable them to play games with their friends and family, but they ultimately wanted to control all aspects of the games.

Overall, we have shown that partial automation can improve the accessibility of digital games to non-gamers and players with motor impairments. Our results suggest several promising ways that partial automation could improve a game's accessibility further, but they should also serve as a warning to designers using partial automation in their games. Even when games inform players of partial automation's activities via awareness cues, automated actions can make players confused about how games are controlled. Players may misunderstand which game actions are under their control and misinterpret stimuli in ways that seem to confirm their erroneous beliefs about what the inputs they provide do in the game. Therefore, partial automation must be designed iteratively with target players to identify and address issues such as automation confusion. The Concordia architecture and associated toolkit, presented in Chapter 8, were designed to facilitate the development of partial automation by providing a template structure that allows developers to easily customize and modify partial automation.

## 9.1 Limitations

The research reported in this thesis provides evidence that partial automation can improve the accessibility of digital games to non-gamers and players with motor impairments. Our study results show that participants were able to play games

with partial automation that they would have been unable to play otherwise. Our measures of disability rely primarily on self-reports. In the rehabilitation gaming study (Chapter 6), we asked participants whether they believed that they could use the gamepad and actively cycle using the viva2. If they did not, then we considered them unable to play. In the automation confusion study (Chapter 7), we specifically recruited non-gamers who had fewer than 100 hours of lifetime video gaming experience and many participants cited great difficulty playing digital games with their friends, siblings, and children. Initially we did not ask participants to play games without the assistance of partial automation because we believed that doing so might negatively impact their experiences of participating in the study and taint their perceptions of partial automation. However, we later decided that some evidence indicating whether participants were able to play without partial automation was needed and so we asked the study's last two participants to play *Spelunky* without assistance. The last participant chose not to play because she believed that she would be unable. The second last participant tried playing *Spelunky* without assistance and ultimately, after much difficulty and frustration, determined that she could not do so. We therefore conclude that partial automation improved the accessibility of these games, as all participants were able to play all of the partially automated games in both studies.

This thesis also reports on participants' experiences of playing games with partial automation; however, these results were based on the feedback of a small number of participants who played a limited selection of partially automated games. In the rehabilitation gaming study (Chapter 6), we analyzed participants' interview data using thematic analysis to generate themes that reflected the experiences of

all of our participants. Each of our six participants shared their own perspective and our themes are intended to capture the commonalities that may generalize to other players. In the automation confusion study (Chapter 7), we analyzed game-play and interview data from ten participants using grounded theory methodology to generate descriptions and explanations for the confused behaviours players exhibited. Each of the concepts and categories we developed in our analysis was drawn from multiple similar observations from different participants and we continued collecting data until we no longer needed to develop new concepts to explain new observations. This stopping mechanism is called *saturation* in grounded theory terminology and it is intended to ensure that grounded theories are generally applicable within the substantive area they are drawn from. Therefore, we believe that the themes presented in this thesis may provide some insight into the potential experiences of other spinal cord injury rehabilitation patients who play *Dino Dash* and *Dozo Quest* with personalizable partial automation and we believe that our grounded theory of automation confusion may help to explain the automation confusion of other non-gamers who play the partially automated *Ninja Showdown* and *Spelunky* games. However, these results were empirically derived and we cannot know for sure how they might generalize to situations that we did not test. We varied the games and forms of partial automation used in our studies, but further investigation is needed to determine how different games or different forms of partial automation might affect players' experiences.

## 9.2 Future Work

While the results presented in this thesis supports our major claim that partial automation can improve the accessibility of digital games, they also suggest several promising ways in which partial automation might improve games accessibility further. Non-gamers wanted partial automation to help them learn how to play without assistance; spinal cord injury rehabilitation patients wanted to play exergames with other patients using their wheelchair controllers; the Concordia architecture and toolkit's capacity to facilitate the development of partial automation could be evaluated by commercial game developers; and partial automation that provides dynamically adjusted levels of assistance could be explored. In this section, we present several potential directions for future research on partial automation.

### 9.2.1 Scaffolding for Non-gamers

While most participants in our automation confusion study believed that partial automation might help them to play the sorts of fast-paced action games their gamer friends and family play, several participants ultimately wanted to be able to play these games without assistance. P2 and P3 believed that they could eventually learn to control all of their avatars' actions but also that doing so would require practice and instruction (e.g., video game courses). Most participants liked that the games' smart avatars lowered the barrier to entry of these games and P2 suggested that partial automation should be designed to scaffold players' learning by incrementally increasing their control over the avatar. She said: "*I want the same [level of control that my kids play with]... Now I can in the [smart] avatar game. If you*

*give me the more options on this I can.*" While we cannot determine from our study data whether the partial automation in *Spelunky* might help novices learn to play without assistance, comparing the performance of non-gamers who play either with or without partial automation that progressively increases players' control in a longitudinal study might indicate whether partial automation can help novices learn complex games. One potential confound, however, is that varying the assistance of partial automation may lead to automation confusion in players.

### 9.2.2 Personalizable Social Gaming in Rehabilitation

Participants in the rehabilitation gaming study said that playing games made exercising more fun and that winning the games made them feel accomplished. P5 and P6 liked being able to personalize the hardware interfaces they used to play the partially automated games and P2 said that playing with others might help patients to get back their "*competitive fire*." For these participants gaming was a beloved pastime that was no longer accessible to them and P2 explained that "[Accessible rehabilitation gaming] *allows you to think that you can still do something that, to be honest, you never thought that you'd be able to do again.*" Since socializing with peers may be important for adjusting to life with spinal cord injury [50], accessible exergaming with others may help patients to form meaningful relationships that enrich their lives. The improved accessibility afforded by partial automation might enable patients with different abilities to play and compete in the same games. A follow-up study in which spinal cord injury rehabilitation patients play partially automated games as a social activity might indicate whether accessible rehabilitation gaming can improve patients' rehabilitation experiences and outcomes.

### 9.2.3 Using the Concordia Toolkit in Industry

In Chapter 8, we evaluated the Concordia toolkit's expressiveness and the potential source code impact of modifying its plug-ins. Our evaluations indicate that the Concordia toolkit may facilitate the development of partial automation, but determining whether it actually can would require further evaluation by target users. In future studies, game developers' impressions of designing and implementing partial automation using the Concordia toolkit could be investigated. Interviews with developers might indicate whether use of the Concordia toolkit could fit within industrial game development processes. Comparing the source code impact of modifying partial automation that was made with or without the Concordia toolkit might indicate whether it can reduce the cost to develop partial automation.

### 9.2.4 Automating the Personalization of Partial Automation

Partial automation is a generally applicable approach to games accessibility that can be realized in radically different ways. In Chapter 3, we described two games accessibility techniques that may use partial automation: player balancing, which assists players with input they can provide, and one-switch games, which reduce the number of inputs players have to provide. These techniques are designed to improve a game's accessibility to different types of disabled players, but both use partial automation and both may be combined to extend a game's accessibility further. It may be that an *interpreted* form of partial automation, as described in Chapter 5, could assess individual players' abilities to provide input and dynamically adjust the assistance provided by partial automation. Further investigation may indicate

whether partial automation such as this can better personalize partial automation to the abilities of individual players.

### 9.3 Summary

We have shown that partial automation is a form of human-AI shared control that can improve the accessibility of digital games to players with motor impairments and non-gamers. Under partial automation, a human player and an AI copilot share control of the game. This creates a tightly-coupled form of human-machine cooperation similar to other uses of shared control in problem domains as diverse as computer-assisted surgery and semi-autonomous driving.

In Chapter 5, we presented a dimension space intended to guide the design of shared control systems, such as partial automation, and in Chapter 6 we showed how the *equal partner* design pattern can be used to improve the accessibility of rehabilitation games. As explained in Chapter 2, people with motor impairments may be unable to provide input to some games, which can limit the accessibility of rehabilitation games involving exercise. In Chapter 6, we demonstrated partial automation’s capacity to personalize rehabilitation games to the abilities of individual players and extend the accessibility afforded by other approaches to games accessibility presented in Chapter 3. While partial automation did make rehabilitation games more accessible to our study participants, it also made two participants frustrated and confused about how the games were controlled. One participant misattributed automated actions to himself, so we set out to investigate how and why this sort of *automation confusion* occurs.

We chose to study non-gamers because we believed them to be susceptible to automation confusion and also because we believed that they may benefit from partial automation. As explained in Chapter 2, non-gamers may have difficulty determining how to respond and providing input in complex games, which may prevent them from playing games socially with their friends and family. In Chapter 7, we showed that partial automation can help non-gamers to play fast-paced action games and we presented a grounded theory of non-gamers' automation confusion in these games. We found that there are many potential sources of automation confusion that can lead non-gamers to develop erroneous mental models, change their attitudes toward the game, and exhibit behaviours that give rise to new sources of confusion. Ultimately, we determined that partial automation can improve games accessibility, but it must be designed iteratively with target players and with consideration for automation confusion. In Chapter 8, we proposed a reference architecture and associated toolkit intended to facilitate the development of partial automation in an iterative design process.

---

## Bibliography

---

- [1] AARSAND, P. A. Computer and Video Games in Family Life: The digital divide as a resource in intergenerational interactions. *Childhood* 14, 2 (May 2007), 235–256. Publisher: SAGE Publications Ltd.
- [2] ABBINK, D. A., CARLSON, T., MULDER, M., DE WINTER, J. C. F., AMINRAVAN, F., GIBO, T. L., AND BOER, E. R. A Topology of Shared Control Systems—Finding Common Ground in Diversity. *IEEE Transactions on Human-Machine Systems* 48, 5 (Oct. 2018), 509–525. Conference Name: IEEE Transactions on Human-Machine Systems.
- [3] ABBINK, D. A., MULDER, M., AND BOER, E. R. Haptic shared control: smoothly shifting control authority? *Cognition, Technology & Work* 14, 1 (Mar. 2012), 19–28.
- [4] ACKERMANN, J. Playing Computer Games as Social Interaction: An Analysis

- of LAN Parties. In *Computer Games and New Media Cultures: A Handbook of Digital Games Studies*, J. Fromme and A. Unger, Eds. Springer Netherlands, Dordrecht, 2012, pp. 465–476.
- [5] ALEXANDER, C. *The timeless way of building*. Oxford University Press, New York, 1979.
- [6] ALEXANDER, C., ISHIKAWA, S., AND SILVERSTEIN, M. *A pattern language: towns, buildings, construction*. Oxford University Press, New York, 1977.
- [7] ALSALEEM, A., IMBURGIA, R., GODINEZ, M., MERRYWEATHER, A., ALTIZER, R., DENNING, T., ROSENBLUTH, J., TRAPP, S., AND WIESE, J. Leveraging Shared Control to Empower People with Tetraplegia to Participate in Extreme Sports. In *The 21st International ACM SIGACCESS Conference on Computers and Accessibility* (Pittsburgh, PA, USA, Oct. 2019), ASSETS ’19, Association for Computing Machinery, pp. 470–481.
- [8] ALSALEEM, A., IMBURGIA, R., MERRYWEATHER, A., ALTIZER, R., ROSENBLUTH, J., TRAPP, S., AND WIESE, J. Experience is not Required: Designing a Sailing Experience for Individuals with Tetraplegia. In *Proceedings of the 2020 ACM on Designing Interactive Systems Conference* (Eindhoven, Netherlands, July 2020), DIS ’20, Association for Computing Machinery, pp. 35–46.
- [9] ALSALEEM, A., IMBURGIA, R., MERRYWEATHER, A., ROSENBLUTH, J., TRAPP, S., AND WIESE, J. Applying Ability-Based Design Principles to Adaptive Outdoor Activities. In *Proceedings of the 2020 ACM on Designing Interactive Systems Conference* (Eindhoven, Netherlands, July 2020), DIS ’20, Association for Computing Machinery, pp. 1–12.

- [10] AMERSHI, S., WELD, D., VORVOREANU, M., FOURNEY, A., NUSHI, B., COLLISSON, P., SUH, J., IQBAL, S., BENNETT, P. N., INKPEN, K., TEEVAN, J., KIKIN-GIL, R., AND HORVITZ, E. Guidelines for Human-AI Interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2019), CHI '19, ACM, pp. 3:1–3:13. event-place: Glasgow, Scotland Uk.
- [11] AROMATARIS, E., AND MUNN, Z., Eds. *JBI Manual for Evidence Synthesis*. JBI, 2020.
- [12] ARTUR, E., AND MINGHIM, R. A novel visual approach for enhanced attribute analysis and selection. *Computers & Graphics* 84 (Nov. 2019), 160–172.
- [13] ATARI. *Home Pong*. Game [Console], 1975. Atari, Sunnyvale, California, USA.
- [14] BAINBRIDGE, L. Ironies of automation. *Automatica* 19, 6 (1983), 775–779.
- [15] BALDWIN, M. S., HIRANO, S., DERAMA, R., MANKOFF, J., AND HAYES, G. R. Blind Navigation on the Water Through Shared Assistive Technology. 4.
- [16] BALTZER, M. C. A., LÓPEZ, D., AND FLEMISCH, F. Towards an interaction pattern language for human machine cooperation and cooperative movement. *Cognition, Technology & Work* 21, 4 (Nov. 2019), 593–606.
- [17] BATEMAN, S., MANDRYK, R. L., STACH, T., AND GUTWIN, C. Target assistance for subtly balancing competitive play. In *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11* (Vancouver, BC, Canada, 2011), ACM Press, p. 2355.

- [18] BEAT GAMES. *Beat Saber*. Game [Oculus VR], May 2019. Beat Games, Prague, Czech Republic.
- [19] BEESTON, J., POWER, C., CAIRNS, P., AND BARLET, M. Accessible Player Experiences (APX): The Players. In *Computers Helping People with Special Needs* (Cham, 2018), K. Miesenberger and G. Kouroupetroglou, Eds., Lecture Notes in Computer Science, Springer International Publishing, pp. 245–253.
- [20] BEMANI. *Dance Dance Revolution*. Game [Arcade], September 1998. Konami, Tokyo, Japan.
- [21] BESNARD, D., GREATHEAD, D., AND BAXTER, G. When mental models go wrong: co-occurrences in dynamic, critical systems. *International Journal of Human-Computer Studies* 60, 1 (Jan. 2004), 117–128.
- [22] BETKER, A., SZTURM, T., AND MOUSSAVI, Z. Development of an Interactive Motivating Tool for Rehabilitation Movements. In *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference* (Jan. 2005), pp. 6893–6896. ISSN: 1558-4615.
- [23] BETKER, A. L., DESAI, A., NETT, C., KAPADIA, N., AND SZTURM, T. Game-based Exercises for Dynamic Short-Sitting Balance Rehabilitation of People With Chronic Spinal Cord and Traumatic Brain Injuries. *Physical Therapy* 87, 10 (Oct. 2007), 1389–1398.
- [24] BEZNOSYK, A., QUAX, P., LAMOTTE, W., AND CONINX, K. The Effect of

- Closely-Coupled Interaction on Player Experience in Casual Games. In *Entertainment Computing - ICEC 2012* (Berlin, Heidelberg, 2012), M. Herrlich, R. Malaka, and M. Masuch, Eds., Lecture Notes in Computer Science, Springer, pp. 243–255.
- [25] BHATIA, S., AND MALHOTRA, J. A survey on impact of lines of code on software complexity. In *2014 International Conference on Advances in Engineering & Technology Research (ICAETR - 2014)* (Aug. 2014), pp. 1–4. ISSN: 2347-9337.
- [26] BIRNBAUM, D., FIEBRINK, R., MALLOCH, J., AND WANDERLEY, M. M. Towards a dimension space for musical devices. In *Proceedings of the 2005 conference on New interfaces for musical expression* (SGP, May 2005), NIME '05, National University of Singapore, pp. 192–195.
- [27] BLUMENBERG, E., PAUL, J., AND PIERCE, G. Travel in the digital age: Vehicle ownership and technology-facilitated accessibility. *Transport Policy* 103 (Mar. 2021), 86–94.
- [28] BORTOLASO, C., BOURDIOL, J., AND GRAHAM, T. C. N. Enhancing Communication and Awareness in Asymmetric Games. In *Entertainment Computing and Serious Games* (Cham, 2019), E. van der Spek, S. Göbel, E. Y.-L. Do, E. Clua, and J. Baalsrud Hauge, Eds., Lecture Notes in Computer Science, Springer International Publishing, pp. 250–262.

- [29] BOURKE, J. A., HAY-SMITH, E. J. C., SNELL, D. L., AND DEJONG, G. Attending to biographical disruption: the experience of rehabilitation following tetraplegia due to spinal cord injury. *Disability and Rehabilitation* 37, 4 (Feb. 2015), 296–303. Publisher: Taylor & Francis eprint: <https://doi.org/10.3109/09638288.2014.918188>.
- [30] BRAUN, V., AND CLARKE, V. Using thematic analysis in psychology. *Qualitative Research in Psychology* 3, 2 (Jan. 2006), 77–101.
- [31] BRAUN, V., AND CLARKE, V. *Successful qualitative research: a practical guide for beginners*. SAGE, Los Angeles, 2013. OCLC: ocn811733656.
- [32] BRAUN, V., AND CLARKE, V. Reflecting on reflexive thematic analysis. *Qualitative Research in Sport, Exercise and Health* 11, 4 (Aug. 2019), 589–597.
- [33] BREWER, R. N., AND KAMESWARAN, V. Understanding the Power of Control in Autonomous Vehicles for People with Vision Impairment. In *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility* (New York, NY, USA, Oct. 2018), ASSETS ’18, Association for Computing Machinery, pp. 185–197.
- [34] BRINKLEY, J., DAILY, S. B., AND GILBERT, J. E. Implementing the ATLAS Self-Driving Vehicle Voice User Interface. Accepted: 2019-05-28T21:25:54Z Publisher: California State University, Northridge.
- [35] BRINKLEY, J., DAILY, S. B., AND GILBERT, J. E. A Policy Proposal to Support Self-Driving Vehicle Accessibility. Accepted: 2019-05-28T21:25:53Z Publisher: California State University, Northridge.

- [36] BRINKLEY, J., GILBERT, J. E., AND DAILY, S. B. A Survey of Visually Impaired Consumers About Self-Driving Vehicles. Accepted: 2018-05-17T22:56:50Z Publisher: California State University, Northridge.
- [37] BRINKLEY, J., HUFF, E. W., POSADAS, B., WOODWARD, J., DAILY, S. B., AND GILBERT, J. E. Exploring the Needs, Preferences, and Concerns of Persons with Visual Impairments Regarding Autonomous Vehicles. *ACM Transactions on Accessible Computing* 13, 1 (Apr. 2020), 3:1–3:34.
- [38] BRINKLEY, J., POSADAS, B., WOODWARD, J., AND GILBERT, J. E. Opinions and Preferences of Blind and Low Vision Consumers Regarding Self-Driving Vehicles: Results of Focus Group Discussions. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility* (New York, NY, USA, Oct. 2017), ASSETS ’17, Association for Computing Machinery, pp. 290–299.
- [39] BROWN, J. A. Digital Gaming Perceptions Among Older Adult Non-gamers. In *Human Aspects of IT for the Aged Population. Applications, Services and Contexts* (Cham, 2017), J. Zhou and G. Salvendy, Eds., Lecture Notes in Computer Science, Springer International Publishing, pp. 217–227.
- [40] BRYANT, A., Ed. *The SAGE handbook of grounded theory*, paperback ed., reprint ed. SAGE, Los Angeles, 2012.
- [41] BURNS, P., KRESSLER, J., AND NASH, M. S. Physiological Responses to Exergaming After Spinal Cord Injury. *Topics in Spinal Cord Injury Rehabilitation* 18, 4 (2012), 331–339.

- [42] BURTON, D. M., DELANEY, A. K., NEWSTEAD, S. V., LOGAN, D., AND FILDES, B. N. *Effectiveness of ABS and Vehicle Stability Control Systems*. Royal Automobile Club of Victoria Ltd, 2004.
- [43] CAPCOM. *Street Fighter 6*. Game [PC], June 2023. Capcom, Osaka, Japan.
- [44] CARD, S. K., MORAN, T. P., AND NEWELL, A., Eds. *The psychology of human-computer interaction*, 1st ed. CRC Press, Boca Raton, 2018. OCLC: 1064714844.
- [45] CARROLL, J. M., AND CARRITHERS, C. Training wheels in a user interface. *Communications of the ACM* 27, 8 (Aug. 1984), 800–806.
- [46] CECHANOWICZ, J. E., GUTWIN, C., BATEMAN, S., MANDRYK, R., AND STAVNESS, I. Improving player balancing in racing games. In *Proceedings of the first ACM SIGCHI annual symposium on Computer-human interaction in play* (New York, NY, USA, Oct. 2014), CHI PLAY '14, Association for Computing Machinery, pp. 47–56.
- [47] CHAMBERS, D. ‘Wii play as a family’: the rise in family-centred video gaming. *Leisure Studies* 31, 1 (Jan. 2012), 69–82. Publisher: Routledge \_eprint: <https://doi.org/10.1080/02614367.2011.568065>.
- [48] CHARMAZ, K. *Constructing grounded theory*, 2nd edition ed. Introducing qualitative methods. Sage, London ; Thousand Oaks, Calif, 2014. OCLC: ocn878133162.
- [49] CHUA, P.-H., JUNG, Y., LWIN, M. O., AND THENG, Y.-L. Let’s play together: Effects of video-game play on intergenerational perceptions among youth

- and elderly participants. *Computers in Human Behavior* 29, 6 (Nov. 2013), 2303–2311.
- [50] CHUN, S., AND LEE, Y. The Role of Leisure in the Experience of Post-traumatic Growth for People with Spinal Cord Injury. *Journal of Leisure Research* 42, 3 (Sept. 2010), 393–415. Publisher: Routledge eprint: <https://doi.org/10.1080/00222216.2010.11950211>.
- [51] CIMOLINO, G., ASKARI, S., AND GRAHAM, T. N. Beyond Fun: Players' Experiences of Accessible Rehabilitation Gaming for Spinal Cord Injury. In *The 23rd International ACM SIGACCESS Conference on Computers and Accessibility* (New York, NY, USA, Oct. 2021), ASSETS '21, Association for Computing Machinery, pp. 1–13.
- [52] CIMOLINO, G., ASKARI, S., AND GRAHAM, T. N. The Role of Partial Automation in Increasing the Accessibility of Digital Games. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play* (New York, NY, USA, Oct. 2021), CHI PLAY '21, ACM.
- [53] CIMOLINO, G., CHEN, R. X., GUTWIN, C., AND GRAHAM, T. N. Automation Confusion: A Grounded Theory of Non-Gamers' Confusion in Partially Automated Action Games. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2023), CHI '23, Association for Computing Machinery, pp. 1–19.
- [54] CIMOLINO, G., AND GRAHAM, T. N. Two Heads Are Better Than One: A Dimension Space for Unifying Human and Artificial Intelligence in Shared Control. In *CHI Conference on Human Factors in Computing Systems* (New

- York, NY, USA, Apr. 2022), CHI '22, Association for Computing Machinery, pp. 1–21.
- [55] CLAWS, T. *The Under Presents*. Game [PC], November 2019. Tender Claws, Los Angeles, USA.
- [56] COLE, T., AND GILLIES, M. Emotional Exploration and the Eudaimonic Gameplay Experience: A Grounded Theory. In *CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2022), CHI '22, Association for Computing Machinery, pp. 1–16.
- [57] COLE, T., AND GILLIES, M. More than a bit of coding: (un-)Grounded (non-)Theory in HCI. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts* (New York, NY, USA, Apr. 2022), CHI EA '22, Association for Computing Machinery, pp. 1–11.
- [58] COLLINS, A., AND GENTNER, D. How people construct mental models. In *Cultural Models in Language and Thought*, D. Holland and N. Quinn, Eds. Cambridge University Press, Cambridge, 1987, pp. 243–266.
- [59] COOK, R. I., POTTER, S. S., WOODS, D. D., AND McDONALD, J. S. Evaluating the human engineering of microprocessor-controlled operating room devices. *Journal of Clinical Monitoring* 7, 3 (July 1991), 217–226.
- [60] COOK, R. I., WOODS, D. D., MCCOLLIGAN, E., AND HOWIE, M. B. Cognitive consequences of clumsy automation on high workload, high consequence human performance. NTRS Author Affiliations: Ohio State Univ. NTRS Document ID: 19910011398 NTRS Research Center: Legacy CDMS (CDMS).

- [61] CUZZORT, S., AND STARNER, T. AstroWheelie: A wheelchair based exercise game. In *2008 12th IEEE International Symposium on Wearable Computers* (Sept. 2008), pp. 113–114. ISSN: 2376-8541.
- [62] DALGLEISH, M. “There are no universal interfaces: how asymmetrical roles and asymmetrical controllers can increase access diversity. *G|A|M|E Games as Art, Media, Entertainment* 1, 7 (2018). Publisher: Ludica Section: Journal.
- [63] DARONNAT, S. Human-Agent Trust Relationships in a Real-Time Collaborative Game. In *Extended Abstracts of the 2020 Annual Symposium on Computer-Human Interaction in Play* (New York, NY, USA, Nov. 2020), CHI PLAY ’20, Association for Computing Machinery, pp. 18–20.
- [64] DARONNAT, S., AZZOPARDI, L., AND HALVEY, M. Impact of agents’ errors on performance, reliance and trust in human-agent collaboration. pp. 1–5.
- [65] DARONNAT, S., AZZOPARDI, L., HALVEY, M., AND DUBIEL, M. Impact of Agent Reliability and Predictability on Trust in Real Time Human-Agent Collaboration. In *Proceedings of the 8th International Conference on Human-Agent Interaction* (New York, NY, USA, Nov. 2020), HAI ’20, Association for Computing Machinery, pp. 131–139.
- [66] DE SCHUTTER, B. Never Too Old to Play: The Appeal of Digital Games to an Older Audience. *Games and Culture* 6, 2 (Mar. 2011), 155–170. Publisher: SAGE Publications.
- [67] DENG, X., YU, Z. L., LIN, C., GU, Z., AND LI, Y. Self-adaptive shared control with brain state evaluation network for human-wheelchair cooperation.

- Journal of Neural Engineering* 17, 4 (July 2020), 045005. Publisher: IOP Publishing.
- [68] DIAZ GANDASEGUI, V. The non-gamer. Accepted: 2011-05-26T15:34:42Z.
- [69] DJI. Mavic Air 2 - Up Your Game, 2021.
- [70] DRAGAN, A. D., LEE, K. C., AND SRINIVASA, S. S. Legibility and predictability of robot motion. In *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)* (Mar. 2013), pp. 301–308. ISSN: 2167-2148.
- [71] DRAGAN, A. D., AND SRINIVASA, S. S. Assistive teleoperation for manipulation tasks. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction* (Boston, Massachusetts, USA, Mar. 2012), HRI ’12, Association for Computing Machinery, pp. 123–124.
- [72] DRAGAN, A. D., AND SRINIVASA, S. S. A policy-blending formalism for shared control. *The International Journal of Robotics Research* 32, 7 (June 2013), 790–805.
- [73] EDEY, J. K., SEABORN, K., BRANJE, C., AND FELS, D. I. Powered to play: A mixed reality game for people driving powered chairs. In *2014 IEEE Games Media Entertainment* (Oct. 2014), pp. 1–8. ISSN: null.
- [74] ELLIS, B. *One-Switch 100*, draft - early version 2 ed. Jan. 2021.
- [75] ENDSLEY, M. R. Toward a Theory of Situation Awareness in Dynamic Systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 37, 1 (Mar. 1995), 32–64.

- [76] ENGEL, J., STURM, J., AND CREMERS, D. Scale-aware navigation of a low-cost quadrocopter with a monocular camera. *Robotics and Autonomous Systems* 62, 11 (Nov. 2014), 1646–1656.
- [77] ENTERTAINMENT SOFTWARE ASSOCIATION OF CANADA. Real Canadian Gamer Essential Facts 2020. Tech. rep., Entertainment Software Association of Canada, 2020.
- [78] ENTERTAINMENT SOFTWARE ASSOCIATION OF CANADA. Bringing Canadians Together through Gaming: Essential Facts 2022. Tech. rep., Entertainment Software Association of Canada, 2022.
- [79] EVANS, R. Varieties of Learning. In *AI Game Programming Wisdom*. Charles River Media, Inc., Rockland, MA, USA, 2002, pp. 567–578.
- [80] EZEH, C., TRAUTMAN, P., DEVIGNE, L., BUREAU, V., BABEL, M., AND CARLSON, T. Probabilistic vs linear blending approaches to shared control for wheelchair driving. In *2017 International Conference on Rehabilitation Robotics (ICORR)* (July 2017), pp. 835–840. ISSN: 1945-7901.
- [81] EZEH, C., TRAUTMAN, P., HOLLOWAY, C., AND CARLSON, T. Comparing shared control approaches for alternative interfaces: A wheelchair simulator experiment. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (Oct. 2017), pp. 93–98.
- [82] FINK, P. D. S., HOLZ, J. A., AND GIUDICE, N. A. Fully Autonomous Vehicles for People with Visual Impairment: Policy, Accessibility, and Future

- Directions. *ACM Transactions on Accessible Computing* 14, 3 (Aug. 2021), 15:1–15:17.
- [83] FITZGERALD, S. G., COOPER, R. A., THORMAN, T., COOPER, R., GUO, S., AND BONINGER, M. L. The Gamecycle Exercise System: Comparison With Standard Ergometry. *The Journal of Spinal Cord Medicine* 27, 5 (Jan. 2004), 453–459.
- [84] FITZGERALD, S. G., COOPER, R. A., ZIPFEL, E., SPAETH, D. M., PUHLMAN, J., KELLEHER, A., COOPER, R., AND GUO, S. The development and preliminary evaluation of a training device for wheelchair users: The GAMEWheels system. *Disability and Rehabilitation: Assistive Technology* 1, 1-2 (Jan. 2006), 129–139.
- [85] FIZZOTTI, G., ROGNONI, C., IMARISIO, A., MENEGHINI, A., PISTARINI, C., AND QUAGLINI, S. Tablet Technology for Rehabilitation after Spinal Cord Injury: a Proof-of-Concept. *Studies in Health Technology and Informatics* 210 (2015), 479–483.
- [86] FLEMISCH, F., ABBINK, D., ITOH, M., AND PACAUX-LEMOINE, M.-P. Special issue on shared and cooperative control. *Cognition, Technology & Work* 21, 4 (Nov. 2019), 553–554.
- [87] FLEMISCH, F., ABBINK, D., ITOH, M., PACAUX-LEMOINE, M.-P., AND WESSEL, G. Shared control is the sharp end of cooperation: Towards a common framework of joint action, shared control and human machine cooperation. *IFAC-PapersOnLine* 49, 19 (Jan. 2016), 72–77.

- [88] FLEMISCH, F., ABBINK, D. A., ITOH, M., PACAUX-LEMOINE, M.-P., AND WESSEL, G. Joining the blunt and the pointy end of the spear: towards a common framework of joint action, human–machine cooperation, cooperative guidance and control, shared, traded and supervisory control. *Cognition, Technology & Work* 21, 4 (Nov. 2019), 555–568.
- [89] FLEMISCH, F., KELSCH, J., LÖPER, C., SCHIEBEN, A., SCHINDLER, J., AND HEESEN, M. Cooperative Control and Active Interfaces for Vehicle Assitsance and Automation.
- [90] FLEMISCH, F. O., BENGLER, K., BUBB, H., WINNER, H., AND BRUDER, R. Towards cooperative guidance and control of highly automated vehicles: H-Mode and Conduct-by-Wire. *Ergonomics* 57, 3 (Mar. 2014), 343–360. Publisher: Taylor & Francis eprint: <https://doi.org/10.1080/00140139.2013.869355>.
- [91] FLEMISCH, F. O. A. The H-Metaphor as a Guideline for Vehicle Automation and Interaction. Tech. rep., Jan. 2003.
- [92] FRANCHI, A., SECCHI, C., RYLL, M., BULTHOFF, H. H., AND GIORDANO, P. R. Shared Control : Balancing Autonomy and Human Assistance with a Group of Quadrotor UAVs. *IEEE Robotics Automation Magazine* 19, 3 (Sept. 2012), 57–68.
- [93] FRISON, A.-K., WINTERSBERGER, P., RIENER, A., AND SCHARTMÜLLER, C. Driving Hotzenplotz: A Hybrid Interface for Vehicle Control Aiming to Maximize Pleasure in Highway Driving. In *Proceedings of the 9th International*

- Conference on Automotive User Interfaces and Interactive Vehicular Applications* (New York, NY, USA, Sept. 2017), AutomotiveUI '17, Association for Computing Machinery, pp. 236–244.
- [94] GAFFURINI, P., BISSOLOTTI, L., CALZA, S., CALABRETTA, C., ORIZIO, C., AND GOBBO, M. Energy metabolism during activity-promoting video games practice in subjects with spinal cord injury: evidences for health promotion. *European Journal of Physical and Rehabilitation Medicine* 49, 1 (Feb. 2013), 23–29.
- [95] GAJADHAR, B., DE KORT, Y., AND IJSSELSTEIJN, W. Influence of social setting on player experience of digital games. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2008), CHI EA '08, Association for Computing Machinery, pp. 3099–3104.
- [96] GAJADHAR, B. J., DE KORT, Y. A. W., AND IJSSELSTEIJN, W. A. Shared Fun Is Doubled Fun: Player Enjoyment as a Function of Social Setting. In *Fun and Games* (Berlin, Heidelberg, 2008), P. Markopoulos, B. de Ruyter, W. IJsselsteijn, and D. Rowland, Eds., Lecture Notes in Computer Science, Springer, pp. 106–117.
- [97] GAME ACCESSIBILITY GUIDELINES. Game accessibility guidelines.
- [98] GARCIA, J. P. P. Games for non-gamers: Approaching video games from a non-gamer perspective. Accepted: 2021-12-20T00:19:11Z.

- [99] GERAVAND, M., WERNER, C., HAUER, K., AND PEER, A. An Integrated Decision Making Approach for Adaptive Shared Control of Mobility Assistance Robots. *International Journal of Social Robotics* 8, 5 (Nov. 2016), 631–648.
- [100] GERLING, K., HICKS, K., KALYN, M., EVANS, A., AND LINEHAN, C. Designing Movement-based Play With Young People Using Powered Wheelchairs. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA, May 2016), CHI ’16, Association for Computing Machinery, pp. 4447–4458.
- [101] GERLING, K., AND MASUCH, M. When gaming is not suitable for everyone: playtesting wii games with frail elderly. In *1st Workshop on Game Accessibility* (2011).
- [102] GERLING, K. M., KALYN, M. R., AND MANDRYK, R. L. KINECTwheels: wheelchair-accessible motion-based game interaction. In *CHI ’13 Extended Abstracts on Human Factors in Computing Systems on - CHI EA ’13* (Paris, France, 2013), ACM Press, p. 3055.
- [103] GERLING, K. M., MANDRYK, R. L., AND KALYN, M. R. Wheelchair-based game design for older adults. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility* (Bellevue, Washington, Oct. 2013), ASSETS ’13, Association for Computing Machinery, pp. 1–8.
- [104] GERLING, K. M., MILLER, M., MANDRYK, R. L., BIRK, M. V., AND SMED-DINCK, J. D. Effects of Balancing for Physical Abilities on Player Performance, Experience and Self-esteem in Exergames. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems* (New York, NY,

- USA, 2014), CHI '14, ACM, pp. 2201–2210. event-place: Toronto, Ontario, Canada.
- [105] GERLING, K. M., SCHULTE, F. P., SMEDDINCK, J., AND MASUCH, M. Game Design for Older Adults: Effects of Age-Related Changes on Structural Elements of Digital Games. In *Entertainment Computing - ICEC 2012* (Berlin, Heidelberg, 2012), M. Herrlich, R. Malaka, and M. Masuch, Eds., Lecture Notes in Computer Science, Springer, pp. 235–242.
- [106] GLASER, B. G., AND STRAUSS, A. L. *The discovery of grounded theory: strategies for qualitative research*, 5. paperback print ed. Aldine Transaction, New Brunswick, 2010.
- [107] GLUCK, A., BOATENG, K., HUFF JR., E. W., AND BRINKLEY, J. Putting Older Adults in the Driver Seat: Using User Enactment to Explore the Design of a Shared Autonomous Vehicle. In *12th International Conference on Automotive User Interfaces and Interactive Vehicular Applications* (New York, NY, USA, Sept. 2020), AutomotiveUI '20, Association for Computing Machinery, pp. 291–300.
- [108] GLUCK, A., HUFF, E. W., ZHANG, M., AND BRINKLEY, J. Lights, Camera, Autonomy! Exploring the Opinions of Older Adults Regarding Autonomous Vehicles Through Enactment. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 64, 1 (Dec. 2020), 1971–1975. Publisher: SAGE Publications Inc.
- [109] GOLDBERG, K., CHEN, B., SOLOMON, R., BUI, S., FARZIN, B., HEITLER, J., POON, D., AND SMITH, G. Collaborative teleoperation via the Internet.

- In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)* (Apr. 2000), vol. 2, pp. 2019–2024 vol.2.
- [110] GOLDBERG, K., DEZHEN SONG, AND LEVANDOWSKI, A. Collaborative teleoperation using networked spatial dynamic voting. *Proceedings of the IEEE* 91, 3 (Mar. 2003), 430–439.
- [111] GOTSIS, M., FRANGOUDES, F., LYMPOURIDIS, V., MANEEKOBKUNWONG, S., TURPIN, D., AND JORDAN-MARSH, M. Skyfarer: a mixed reality shoulder exercise game. In *ACM SIGGRAPH 2013 Studio Talks on - SIGGRAPH '13* (Anaheim, California, 2013), ACM Press, pp. 1–1.
- [112] GOTSIS, M., LYMPOURIDIS, V., REQUEJO, P., HAUBERT, L. L., Poulos, I. C., FRANGOUDES, F., TURPIN, D., AND JORDAN-MARSH, M. Skyfarer: Design Case Study of a Mixed Reality Rehabilitation Video Game. In *International Conference of Design, User Experience, and Usability* (2014), Springer, pp. 699–710.
- [113] GOTSIS, M., TASSE, A., SWIDER, M., LYMPOURIDIS, V., Poulos, I. C., THIN, A. G., TURPIN, D., TUCKER, D., AND JORDAN-MARSH, M. Mixed reality game prototypes for upper body exercise and rehabilitation. In *2012 IEEE Virtual Reality Workshops (VRW)* (Mar. 2012), pp. 181–182. ISSN: 1087-8270.
- [114] GRAF, R., BENAWRI, P., WHITESALL, A. E., CARICHNER, D., LI, Z., NEBELING, M., AND KIM, H. S. iGYM: An Interactive Floor Projection System for Inclusive Exergame Environments. In *Proceedings of the Annual Symposium*

- on Computer-Human Interaction in Play* (Barcelona, Spain, Oct. 2019), CHI PLAY '19, Association for Computing Machinery, pp. 31–43.
- [115] GRAHAM, T. C. N., WATTS, L. A., CALVARY, G., COUTAZ, J., DUBOIS, E., AND NIGAY, L. A dimension space for the design of interactive systems within their physical environments. In *Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques* (New York, NY, USA, Aug. 2000), DIS '00, Association for Computing Machinery, pp. 406–416.
- [116] GUO, S., GRINDLE, G., AUTHIER, E., COOPER, R., FITZGERALD, S., KELLEHER, A., AND COOPER, R. Development and qualitative assessment of the GAME<sup>Cycle</sup> exercise system. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 14, 1 (Mar. 2006), 83–90.
- [117] GURNEL, H., MARCHAL, M., LAUNAY, L., BEUZIT, L., AND KRUPA, A. Design of haptic guides for pre-positioning assistance of a comanipulated needle. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)* (Oct. 2019), pp. 478–485. ISSN: 2577-1655.
- [118] GUTIERREZ, M. A., D'HARO, L. F., AND BANCHS, R. A Multimodal Control Architecture for Autonomous Unmanned Aerial Vehicles. In *Proceedings of the Fourth International Conference on Human Agent Interaction* (New York, NY, USA, Oct. 2016), HAI '16, Association for Computing Machinery, pp. 107–110.

- [119] GUTWIN, C., AND GREENBERG, S. A Descriptive Framework of Workspace Awareness for Real-Time Groupware. *Computer Supported Cooperative Work (CSCW)* 11, 3-4 (Sept. 2002), 411–446.
- [120] GUTWIN, C., AND GREENBERG, S. The importance of awareness for team cognition in distributed collaboration. In *Team cognition: Understanding the factors that drive process and performance*. American Psychological Association, Washington, DC, US, 2004, pp. 177–201.
- [121] GUTWIN, C., GREENBERG, S., AND ROSEMAN, M. Workspace Awareness in Real-Time Distributed Groupware: Framework, Widgets, and Evaluation. In *Proceedings of HCI on People and Computers XI* (Berlin, Heidelberg, Jan. 1996), HCI '96, Springer-Verlag, pp. 281–298.
- [122] GUTWIN, C., ROSEMAN, M., AND GREENBERG, S. A usability study of awareness widgets in a shared workspace groupware system. In *Proceedings of the 1996 ACM conference on Computer supported cooperative work* (Boston, Massachusetts, USA, Nov. 1996), CSCW '96, Association for Computing Machinery, pp. 258–267.
- [123] HARNESS, E. T., YOZBATIRAN, N., AND CRAMER, S. C. Effects of intense exercise in chronic spinal cord injury. *Spinal Cord* 46, 11 (Nov. 2008), 733–737. Number: 11 Publisher: Nature Publishing Group.
- [124] HATTWICK, I., AND WANDERLEY, M. A dimension space for evaluating collaborative musical performance systems. In *Proceedings of the International Conference on New Interfaces for Musical Expression* (Ann Arbor, Michigan, 2012), University of Michigan.

- [125] HENRIKSSON, P., AND PETERS, B. Safety and mobility of people with disabilities driving adapted cars. *Scandinavian Journal of Occupational Therapy* 11, 2 (June 2004), 54–61. Publisher: Taylor & Francis \_eprint: <https://doi.org/10.1080/11038120410020511>.
- [126] HERNANDEZ, H. A., GRAHAM, T. N., FEHLINGS, D., SWITZER, L., YE, Z., BELLAY, Q., HAMZA, M. A., SAVERY, C., AND STACH, T. Design of an Exergaming Station for Children with Cerebral Palsy. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2012), CHI '12, ACM, pp. 2619–2628.
- [127] HERNANDEZ, H. A., YE, Z., GRAHAM, T. N., FEHLINGS, D., AND SWITZER, L. Designing Action-based Exergames for Children with Cerebral Palsy. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2013), CHI '13, ACM, pp. 1261–1270.
- [128] HICKS, A. L., MARTIN GINIS, K. A., PELLETIER, C. A., DITOR, D. S., FOULON, B., AND WOLFE, D. L. The effects of exercise training on physical capacity, strength, body composition and functional performance among adults with spinal cord injury: a systematic review. *Spinal Cord* 49, 11 (Nov. 2011), 1103–1127.
- [129] HICKS, K., AND GERLING, K. Exploring Casual Exergames with Kids Using Wheelchairs. In *Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play* (London, United Kingdom, Oct. 2015), CHI PLAY '15, Association for Computing Machinery, pp. 541–546.

- [130] HOUGAARD, B. I., ROSSAU, I. G., CZAPLA, J. J., MIKÓ, M. A., SKAMMELSEN, R. B., KNOCHE, H., AND JOCHUMSEN, M. Who Willed It? Decreasing Frustration by Manipulating Perceived Control through Fabricated Input for Stroke Rehabilitation BCI Games. *Proceedings of the ACM on Human-Computer Interaction 5*, CHI PLAY (Oct. 2021), 235:1–235:19.
- [131] HUFF, E. W., DELLMARIA, N., POSADAS, B., AND BRINKLEY, J. Am I Too Old to Drive? Opinions of Older Adults on Self-Driving Vehicles. In *The 21st International ACM SIGACCESS Conference on Computers and Accessibility* (New York, NY, USA, Oct. 2019), ASSETS ’19, Association for Computing Machinery, pp. 500–509.
- [132] HUFF, E. W., ZHANG, M., AND BRINKLEY, J. Enacting into Reality: Using User Enactment to Explore the Future of Autonomous Vehicle Design. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting 64*, 1 (Dec. 2020), 1561–1565. Publisher: SAGE Publications Inc.
- [133] HUNICKE, R. The case for dynamic difficulty adjustment in games. In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology* (New York, NY, USA, June 2005), ACE ’05, Association for Computing Machinery, pp. 429–433.
- [134] HWANG, S., SCHNEIDER, A. L. J., CLARKE, D., MACINTOSH, A., SWITZER, L., FEHLINGS, D., AND GRAHAM, T. N. How Game Balancing Affects Play: Player Adaptation in an Exergame for Children with Cerebral Palsy. In *Proceedings of the 2017 Conference on Designing Interactive Systems - DIS ’17* (Edinburgh, United Kingdom, 2017), ACM Press, pp. 699–710.

- [135] HÄUSLSCHMID, R., VON BÜLOW, M., PFLEGING, B., AND BUTZ, A. Supporting Trust in Autonomous Driving. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces* (New York, NY, USA, Mar. 2017), IUI '17, Association for Computing Machinery, pp. 319–329.
- [136] IACOPETTI, F., FANUCCI, L., RONCELLA, R., GIUSTI, D., AND SCEBBA, A. Game Console Controller Interface for People with Disability. In *2008 International Conference on Complex, Intelligent and Software Intensive Systems* (Mar. 2008), pp. 757–762.
- [137] INAGAKI, T. Adaptive automation: Sharing and trading of control. In *Handbook of cognitive task design*, Human factors and ergonomics. Lawrence Erlbaum Associates Publishers, Mahwah, NJ, US, 2003, pp. 147–169.
- [138] INTELLIGENT SYSTEMS. *Fire Emblem: Three Houses*. Game [Switch], July 2019. Intelligent Systems, Kyoto, Japan.
- [139] IZADI, V., YERAVDEKAR, A., AND GHASEMI, A. Determination of Roles and Interaction Modes in a Haptic Shared Control Framework. American Society of Mechanical Engineers Digital Collection.
- [140] JACOBS, P. L., AND NASH, M. S. Exercise Recommendations for Individuals with Spinal Cord Injury. *Sports Medicine* 34, 11 (Sept. 2004), 727–751.
- [141] JAKOPEC, M., HARRIS, S. J., RODRIGUEZ Y BAENA, F., GOMES, P., COBB, J., AND DAVIES, B. L. Preliminary Results of an Early Clinical Experience with the Acrobot™ System for Total Knee Replacement Surgery. In *Medical Image Computing and Computer-Assisted Intervention — MICCAI 2002* (Berlin,

- Heidelberg, 2002), T. Dohi and R. Kikinis, Eds., Lecture Notes in Computer Science, Springer, pp. 256–263.
- [142] JANINE, F., FULLERTON, T., JACQUELYN FORD, M., AND PEARCE, C. The hegemony of play. In *DiGRA '07 - Proceedings of the 2007 DiGRA International Conference: Situated Play* (September 2007), The University of Tokyo.
- [143] JANSSEN, C. P., DONKER, S. F., BRUMBY, D. P., AND KUN, A. L. History and future of human-automation interaction. *International Journal of Human-Computer Studies* 131 (Nov. 2019), 99–107.
- [144] JANSZ, J., AND MARTENS, L. Gaming at a LAN event: the social context of playing video games. *New Media & Society* 7, 3 (June 2005), 333–355. Publisher: SAGE Publications.
- [145] JARAMILLO, J. P., JOHANSON, M. E., AND KIRATLI, B. J. Upper limb muscle activation during sports video gaming of persons with spinal cord injury. *The Journal of Spinal Cord Medicine* 42, 1 (Jan. 2019), 77–85.
- [146] JENSEN, M. M., AND GRØNBÆK, K. Design Strategies for Balancing Exertion Games: A Study of Three Approaches. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems - DIS '16* (Brisbane, QLD, Australia, 2016), ACM Press, pp. 936–946.
- [147] JIANG, J., AND ASTOLFI, A. Shared-control for a UAV operating in the 3D space. In *2015 European Control Conference (ECC)* (July 2015), pp. 1633–1638.

- [148] JIANG, S. D., AND ODOM, J. Toward Initiative Decision-Making for Distributed Human-Robot Teams. In *Proceedings of the 6th International Conference on Human-Agent Interaction* (New York, NY, USA, Dec. 2018), HAI '18, Association for Computing Machinery, pp. 286–292.
- [149] JOHNS, M., MOK, B., SIRKIN, D., GOWDA, N., SMITH, C., TALAMONTI, W., AND JU, W. Exploring shared control in automated driving. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)* (Mar. 2016), pp. 91–98. ISSN: 2167-2148.
- [150] JOHNSON-LAIRD, P. *How We Reason*. Oxford University Press, Oct. 2008.
- [151] KANE, G., EGGLERS, G., BOESECKE, R., RACZKOWSKY, J., WÖRN, H., MAR-MULLA, R., AND MÜHLING, J. System Design of a Hand-Held Mobile Robot for Craniotomy. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2009* (Berlin, Heidelberg, 2009), G.-Z. Yang, D. Hawkes, D. Rueckert, A. Noble, and C. Taylor, Eds., Lecture Notes in Computer Science, Springer, pp. 402–409.
- [152] KAPOOR, A., KUMAR, R., AND TAYLOR, R. H. Simple Biomanipulation Tasks with “Steady Hand” Cooperative Manipulator. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2003* (Berlin, Heidelberg, 2003), R. E. Ellis and T. M. Peters, Eds., Lecture Notes in Computer Science, Springer, pp. 141–148.
- [153] KAYE, L. K., AND BRYCE, J. Putting the ”Fun Factor” Into Gaming: The Influence of Social Contexts on Experiences of Playing Videogames.

- [154] KELLEHER, C., AND PAUSCH, R. Stencils-based tutorials: design and evaluation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2005), CHI '05, Association for Computing Machinery, pp. 541–550.
- [155] KERN, D., STRINGER, M., FITZPATRICK, G., AND SCHMIDT, A. Curball-A Prototype Tangible Game for Inter-Generational Play. In *15th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'06)* (June 2006), pp. 412–418. ISSN: 1524-4547.
- [156] KHOO, E. T., CHEOK, A. D., NGUYEN, T. H. D., AND PAN, Z. Age invaders: social and physical inter-generational mixed reality family entertainment. *Virtual Reality* 12, 1 (Mar. 2008), 3–16.
- [157] KHOO, E. T., MERRITT, T., AND CHEOK, A. D. Designing a Mixed Reality Intergenerational Entertainment System. In *The Engineering of Mixed Reality Systems*, E. Dubois, P. Gray, and L. Nigay, Eds., Human-Computer Interaction Series. Springer, London, 2010, pp. 121–141.
- [158] KIANZAD, S., HUANG, Y., XIAO, R., AND MACLEAN, K. E. Phasking on Paper: Accessing a Continuum of PHysically Assisted SKetchING. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2020), CHI '20, Association for Computing Machinery, pp. 1–12.
- [159] KIM, H., KIM, S., LEE, B., PAK, J., SOHN, M., LEE, G., AND LEE, W. Digital rubbing: playful and intuitive interaction technique for transferring a graphic image onto paper with pen-based computing. In *CHI '08 Extended*

- Abstracts on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2008), CHI EA '08, Association for Computing Machinery, pp. 2337–2342.
- [160] KIRMAN, B., LINEHAN, C., AND FELTWELL, T. Inscrutable Games: How Players Respond to Illegible or Opaque Game Design. In *25th International Academic Mindtrek conference* (New York, NY, USA, Nov. 2022), Academic Mindtrek 2022, Association for Computing Machinery, pp. 31–40.
- [161] KIRSHBLUM, S. C., BURNS, S. P., BIERING-SORENSEN, F., DONOVAN, W., GRAVES, D. E., JHA, A., JOHANSEN, M., JONES, L., KRASSIOUKOV, A., MULCAHEY, M., SCHMIDT-READ, M., AND WARING, W. International standards for neurological classification of spinal cord injury (Revised 2011). *The Journal of Spinal Cord Medicine* 34, 6 (Nov. 2011), 535–546.
- [162] KIZONY, R., RAZ, L., KATZ, N., WEINGARDEN, H., AND WEISS, P. L. T. Video-capture virtual reality system for patients with paraplegic spinal cord injury. *The Journal of Rehabilitation Research and Development* 42, 5 (2005), 595.
- [163] KLAPPERICH, H., AND HASSENZAHL, M. Hotzenplotz: Reconciling Automation with Experience. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction* (New York, NY, USA, Oct. 2016), NordiCHI '16, Association for Computing Machinery, pp. 1–10.
- [164] KLAYMAN, J., AND HA, Y.-W. Hypothesis testing in rule discovery: Strategy, structure, and content. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 15 (1989), 596–604. Place: US Publisher: American Psychological Association.

- [165] KONAMI. *Frogger*. Game [Arcade], June 1981. Konami, Tokyo, Japan.
- [166] KONAMI DIGITAL ENTERTAINMENT. *ZONE OF THE ENDERS THE 2nd RUNNER : MARS*. Game [PC], September 2018. Konami Digital Entertainment, Tokyo, Japan.
- [167] KOO, J., KWAC, J., JU, W., STEINERT, M., LEIFER, L., AND NASS, C. Why did my car just do that? Explaining semi-autonomous driving actions to improve driver understanding, trust, and performance. *International Journal on Interactive Design and Manufacturing (IJIDeM)* 9, 4 (Nov. 2015), 269–275.
- [168] KOOPMAN, P. A Case Study of Toyota Unintended Acceleration and Software Safety, Oct. 2014.
- [169] KOSTER, R. *A theory of fun for game design*. Paraglyph Press, Scottsdale, AZ, 2005. OCLC: ocm57406861.
- [170] KOSTER, R. Practical Creativity, 2014.
- [171] KOW, Y. M., WEN, J., AND CHEN, Y. Designing online games for real-life relationships: examining QQ farm in intergenerational play. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work* (New York, NY, USA, Feb. 2012), CSCW ’12, Association for Computing Machinery, pp. 613–616.
- [172] KULESZA, T., STUMPF, S., BURNETT, M., AND KWAN, I. Tell me more? the effects of mental model soundness on personalizing an intelligent agent. In

- Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Austin, Texas, USA, May 2012), CHI '12, Association for Computing Machinery, pp. 1–10.
- [173] KUTNER, L. A., OLSON, C. K., WARNER, D. E., AND HERTZOG, S. M. Parents' and Sons' Perspectives on Video Game Play: A Qualitative Study. *Journal of Adolescent Research* 23, 1 (Jan. 2008), 76–96. Publisher: SAGE Publications Inc.
- [174] KÖRBER, M., BASELER, E., AND BENGLER, K. Introduction matters: Manipulating trust in automation and reliance in automated driving. *Applied Ergonomics* 66 (Jan. 2018), 18–31.
- [175] LANGER, E. J. The illusion of control. *Journal of Personality and Social Psychology* 32, 2 (1975), 311–328. Place: US Publisher: American Psychological Association.
- [176] LASECKI, W., MILLER, C., SADILEK, A., ABUMOUSSA, A., BORRELLO, D., KUSHALNAGAR, R., AND BIGHAM, J. Real-time Captioning by Groups of Non-experts. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2012), UIST '12, ACM, pp. 23–34. event-place: Cambridge, Massachusetts, USA.
- [177] LASECKI, W. S., MILLER, C. D., AND BIGHAM, J. P. Warping Time for More Effective Real-time Crowdsourcing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2013), CHI '13, ACM, pp. 2033–2036. event-place: Paris, France.

- [178] LASECKI, W. S., MURRAY, K. I., WHITE, S., MILLER, R. C., AND BIGHAM, J. P. Real-time crowd control of existing interfaces. In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (Santa Barbara, California, USA, Oct. 2011), UIST '11, Association for Computing Machinery, pp. 23–32.
- [179] LEE, J. D., AND SEE, K. A. Trust in Automation: Designing for Appropriate Reliance. *Human Factors* 46, 1 (Mar. 2004), 50–80.
- [180] LEE, J. M., AND JU, D. Y. Classification of Human-Vehicle Interaction: User Perspectives on Design. *Social Behavior and Personality: an international journal* 46, 7 (July 2018), 1057–1070.
- [181] LEMOINE, M. P., DEBERNARD, S., CREVITS, I., AND MILLOT, P. Cooperation between humans and machines: First results of an experiment with a multi-level cooperative organisation in air traffic control. *Computer Supported Cooperative Work (CSCW)* 5, 2 (June 1996), 299–321.
- [182] LEVESON, N., AND TURNER, C. An investigation of the Therac-25 accidents. *Computer* 26, 7 (July 1993), 18–41. Conference Name: Computer.
- [183] LI, B., SUN, X., LEUNG, H., AND ZHANG, S. A survey of code-based change impact analysis techniques. *Software Testing, Verification and Reliability* 23, 8 (2013), 613–646. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/stvr.1475>.

- [184] LI, Q., CHEN, W., AND WANG, J. Dynamic shared control for human-wheelchair cooperation. In *2011 IEEE International Conference on Robotics and Automation* (May 2011), pp. 4278–4283. ISSN: 1050-4729.
- [185] LID, I. M. Universal Design and disability: an interdisciplinary perspective. *Disability and Rehabilitation* 36, 16 (Aug. 2014), 1344–1349. Publisher: Taylor & Francis eprint: <https://doi.org/10.3109/09638288.2014.931472>.
- [186] LINDBERG, J., KREUTER, M., TAFT, C., AND PERSON, L.-O. Patient participation in care and rehabilitation from the perspective of patients with spinal cord injury. *Spinal Cord* 51, 11 (Nov. 2013), 834–837. Number: 11 Publisher: Nature Publishing Group.
- [187] LOPAREV, A., LASECKI, W. S., MURRAY, K. I., AND BIGHAM, J. P. Introducing shared character control to existing video games. In *Proceedings of the 9th International Conference on the Foundations of Digital Games, FDG 2014, Liberty of the Seas, Caribbean, April 3-7, 2014* (2014), M. Mateas, T. Barnes, and I. Bogost, Eds., Society for the Advancement of the Science of Digital Games.
- [188] LOUPES, A. C., NUNES, U., VAZ, L., AND VAZ, L. Assisted navigation based on shared-control, using discrete and sparse human-machine interfaces. In *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology* (Aug. 2010), pp. 471–474. ISSN: 1558-4615.
- [189] LOUPES, A. C., PIRES, G., VAZ, L., AND NUNES, U. Wheelchair navigation assisted by Human-Machine shared-control and a P300-based Brain Computer

- Interface. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Sept. 2011), pp. 2438–2444. ISSN: 2153-0866.
- [190] LOPES, P., ION, A., MUELLER, W., HOFFMANN, D., JONELL, P., AND BAUDISCH, P. Proprioceptive Interaction. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2015), CHI ’15, Association for Computing Machinery, pp. 939–948.
- [191] LOPES, P., YÜKSEL, D., GUIMBRETIÈRE, F., AND BAUDISCH, P. Muscle-plotter: An Interactive System based on Electrical Muscle Stimulation that Produces Spatial Output. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (New York, NY, USA, Oct. 2016), UIST ’16, Association for Computing Machinery, pp. 207–217.
- [192] MACLACHLAN, R. A., BECKER, B. C., TABARES, J. C., PODNAR, G. W., LOBES, L. A., AND RIVIERE, C. N. Micron: An Actively Stabilized Handheld Tool for Microsurgery. *IEEE Transactions on Robotics* 28, 1 (Feb. 2012), 195–212. Conference Name: IEEE Transactions on Robotics.
- [193] MACLEAN, A., AND MCKERLIE, D. Design space analysis and use representations. In *Scenario-based design: envisioning work and technology in system development*. John Wiley & Sons, Inc., USA, Sept. 1995, pp. 183–207.
- [194] MAHMUD, A. A., MUBIN, O., SHAHID, S., AND MARTENS, J.-B. Designing social games for children and older adults: Two related case studies. *Entertainment Computing* 1, 3 (Dec. 2010), 147–156.

- [195] MALONE, L. A., PADALABALANARAYANAN, S., MCCROSKEY, J., AND THIRUMALAI, M. Assessment of Active Video Gaming Using Adapted Controllers by Individuals With Physical Disabilities: A Protocol. *JMIR Research Protocols* 6, 6 (June 2017), e116.
- [196] MALONE, L. A., THIRUMALAI, M., PADALABALANARAYANAN, S., NEAL, W. N., BOWMAN, S., AND MEHTA, T. Energy Expenditure and Enjoyment During Active Video Gaming Using an Adapted Wii Fit Balance Board in Adults with Physical Disabilities: Observational Study. *JMIR Serious Games* 7, 1 (Feb. 2019), e11326.
- [197] MASON, L., GERLING, K., DICKINSON, P., AND HOLOPAINEN, J. Dash Lane: An Adaptive Exergame for People Using Manual Wheelchairs. In *Companion Publication of the 2020 ACM Designing Interactive Systems Conference* (Eindhoven, Netherlands, July 2020), DIS' 20 Companion, Association for Computing Machinery, pp. 321–324.
- [198] MASON, L., GERLING, K., DICKINSON, P., AND HOLOPAINEN, J. Dash Lane Mobile: Exploring Hypercasual Play to Provide Accessible Physically Active Breaks. In *Extended Abstracts of the 2020 Annual Symposium on Computer-Human Interaction in Play* (New York, NY, USA, Nov. 2020), CHI PLAY '20, Association for Computing Machinery, pp. 311–315.
- [199] MAT ROSLY, M., HALAKI, M., MAT ROSLY, H., CUESTA, V., HASNAN, N., DAVIS, G. M., AND HUSAIN, R. Exergaming for Individuals with Spinal Cord Injury: A Pilot Study. *Games for Health Journal* 6, 5 (Oct. 2017), 279–289.

- [200] MAT ROSLY, M., MAT ROSLY, H., HASNAN, N., DAVIS, G. M., AND HUSAIN, R. Exergaming boxing versus heavy-bag boxing: are these equipotent for individuals with spinal cord injury? *European Journal of Physical and Rehabilitation Medicine* 53, 4 (Aug. 2017), 527–534.
- [201] MAT ROSLY, M., MAT ROSLY, H., OAM, G. M. D., HUSAIN, R., AND HASNAN, N. Exergaming for individuals with neurological disability: a systematic review. *Disability and Rehabilitation* 39, 8 (Apr. 2017), 727–735.
- [202] MAT ROSLY, M., MAT ROSLY, H., OAM, G. M. D., HUSAIN, R., AND HASNAN, N. Exergaming for individuals with neurological disability: a systematic review. *Disability and Rehabilitation* 39, 8 (Apr. 2017), 727–735.
- [203] MEDEIROS, L., AND COUTINHO, F. Developing an Accessible One-Switch Game for Motor Impaired Players. In *Proceedings of SBGames 2015* (Teresina, PI, Brazil, Nov. 2015), pp. 236–239.
- [204] MICHON, J. A. A Critical View of Driver Behavior Models: What Do We Know, What Should We Do? In *Human Behavior and Traffic Safety*, L. Evans and R. C. Schwing, Eds. Springer US, Boston, MA, 1985, pp. 485–524.
- [205] MILLINGTON, I., FUNGE, J., AND FUNGE, J. *Artificial Intelligence for Games*. CRC Press, Dec. 2018.
- [206] MILLOT, P., AND KAMOUN, A. An Implicit Method for Dynamic Task Allocation Between Man and Computer in Supervision Posts of Automated Processes. *IFAC Proceedings Volumes* 21, 5 (June 1988), 77–82.

- [207] MILLOT, P., AND PACAUX-LEMOINE, M.-P. A Common Work Space for a mutual enrichment of Human-machine Cooperation and Team-Situation Awareness. *IFAC Proceedings Volumes 46*, 15 (Jan. 2013), 387–394.
- [208] MNIIH, V., KAVUKCUOGLU, K., SILVER, D., RUSU, A. A., VENESS, J., BELLEMARE, M. G., GRAVES, A., RIEDMILLER, M., FIDJELAND, A. K., OSTROVSKI, G., PETERSEN, S., BEATTIE, C., SADIK, A., ANTONOGLOU, I., KING, H., KUMARAN, D., WIERSTRA, D., LEGG, S., AND HASSABIS, D. Human-level control through deep reinforcement learning, Feb. 2015.
- [209] MOSSMOUTH. *Spelunky*. Game [PC], February 2008. Mossmouth.
- [210] MUELLER, F., KHOT, R. A., GERLING, K., AND MANDRYK, R. Exertion Games. *Foundations and Trends® in Human–Computer Interaction* 10, 1 (2016), 1–86.
- [211] MUELLER, F., VETERE, F., GIBBS, M., EDGE, D., AGAMANOLIS, S., SHERIDAN, J., AND HEER, J. Balancing exertion experiences. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12* (Austin, Texas, USA, 2012), ACM Press, p. 1853.
- [212] MULROY, S. J., THOMPSON, L., KEMP, B., HATCHETT, P. P., NEWSAM, C. J., LUPOLD, D. G., HAUBERT, L. L., EBERLY, V., GE, T.-T., AZEN, S. P., WINSTEIN, C. J., GORDON, J., AND PHYSICAL THERAPY CLINICAL RESEARCH NETWORK (PTCLINRESNET). Strengthening and optimal movements for painful shoulders (STOMPS) in chronic spinal cord injury: a randomized controlled trial. *Physical Therapy* 91, 3 (Mar. 2011), 305–324.

- [213] MUSIĆ, S., AND HIRCHE, S. Classification of human-robot team interaction paradigms. *IFAC-PapersOnLine* 49, 32 (Jan. 2016), 42–47.
- [214] MUSIĆ, S., AND HIRCHE, S. Control sharing in human-robot team interaction. *Annual Reviews in Control* 44 (Jan. 2017), 342–354.
- [215] NAS, K., YAZMALAR, L., ŞAH, V., AYDIN, A., AND ÖNEŞ, K. Rehabilitation of spinal cord injuries. *World Journal of Orthopedics* 6, 1 (Jan. 2015), 8–16.
- [216] NGUYEN, V., DEEDS-RUBIN, S., TAN, T., AND BOEHM, B. A SLOC counting standard. In *Cocomo ii forum* (2007), vol. 2007, Citeseer, pp. 1–16.
- [217] NINTENDO EAD. *Wii Sports*. Game [Wii], November 2006. Nintendo EAD, Kyoto, Japan.
- [218] NINTENDO EAD. *Wii Fit*. Game [Wii], December 2007. Nintendo EAD, Kyoto, Japan.
- [219] NINTENDO EAD. *Mario Kart 8 Deluxe*. Game [Nintendo Switch], April 2017. Nintendo, Kyoto, Japan.
- [220] NORMAN, D. A. *The design of everyday things*, revised and expanded edition ed. Basic Books, New York, New York, 2013.
- [221] NORMAN, D. A. *Some Observations on Mental Models*. Psychology Press, Jan. 2014. Pages: 15-22 Publication Title: Mental Models.
- [222] NORMAN, S. D., AND ORLADY, H. W. *Flight deck automation: Promises and realities*. Sept. 1989. NTRS Author Affiliations: NASA Ames Research Center, Orlady Associates, Inc., Los Gatos NTRS Meeting Information: Proceedings

- of a NASA/FAA/Industry Workshop; 1988-08-01 to 1988-08-04; undefined NTRS Report/Patent Number: A-89196 NTRS Document ID: 19900004068 NTRS Research Center: Legacy CDMS (CDMS).
- [223] NOVA, N., MIYAKE, K., CHIU, W., AND KWON, N. *Curious Rituals: Gestural Interactionin the Digital Everyday*. Sept. 2012.
- [224] O'CONNOR, T. J., COOPER, R. A., FITZGERALD, S. G., DVORZNAK, M. J., BONINGER, M. L., VANSICKLE, D. P., AND GLASS, L. Evaluation of a Manual Wheelchair Interface to Computer Games. *Neurorehabilitation and Neural Repair* 14, 1 (Mar. 2000), 21–31.
- [225] O'CONNOR, T. J., FITZGERALD, S. G., COOPER, R. A., THORMAN, T. A., AND BONINGER, M. L. Does computer game play aid in motivation of exercise and increase metabolic activity during wheelchair ergometry? *Medical Engineering & Physics* 23, 4 (May 2001), 267–273.
- [226] O'CONNOR, T. J., FITZGERALD, S. G., COOPER, R. A., THORMAN, T. A., AND BONINGER, M. L. Kinetic and physiological analysis of the GAME(Wheels) system. *Journal of Rehabilitation Research and Development* 39, 6 (Dec. 2002), 627–634.
- [227] OH, Y., WU, S.-W., TOUSSAINT, M., AND MAINPRICE, J. Natural Gradient Shared Control. In *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)* (Aug. 2020), pp. 1223–1229. ISSN: 1944-9437.

- [228] ON-ROAD AUTOMATED DRIVING (ORAD) COMMITTEE. Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. Standard J3016\_202104, SAE International, Apr. 2021.
- [229] ONEY, S., LUNDGARD, A., KROSNICK, R., NEBELING, M., AND LASECKI, W. S. Arboretum and Arbility: Improving Web Accessibility Through a Shared Browsing Architecture. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (Berlin, Germany, Oct. 2018), UIST '18, Association for Computing Machinery, pp. 937–949.
- [230] OSMANOVIC, S., AND PECCHIONI, L. Beyond Entertainment: Motivations and Outcomes of Video Game Playing by Older Adults and Their Younger Family Members. *Games and Culture* 11, 1-2 (Jan. 2016), 130–149. Publisher: SAGE Publications.
- [231] OSSMANN, R., THALLER, D., NUSSBAUM, G., VEIGL, C., AND WEISS, C. Making the playstation 3 accessible with asterics. In *Computers Helping People with Special Needs* (Berlin, Heidelberg, 2012), K. Miesenberger, A. Karshmer, P. Penaz, and W. Zagler, Eds., Springer Berlin Heidelberg, pp. 443–450.
- [232] PACAUX-LEMOINE, M. P., AND DEBERNARD, S. A Common Work Space to Support the Air Traffic Control. *IFAC Proceedings Volumes* 33, 12 (June 2000), 75–78.
- [233] PACAUX-LEMOINE, M.-P., AND FLEMISCH, F. Layers of shared and cooperative control, assistance, and automation. *Cognition, Technology & Work* 21, 4 (Nov. 2019), 579–591.

- [234] PACAUX-LEMOINE, M.-P., AND MAKOTO, I. Towards Vertical and Horizontal Extension of Shared Control Concept. In *2015 IEEE International Conference on Systems, Man, and Cybernetics* (Oct. 2015), pp. 3086–3091.
- [235] PALMER, C. The Boeing 737 Max Saga: Automating Failure. *Engineering* 6, 1 (Feb. 2020), 2–3.
- [236] PARASURAMAN, R., AND RILEY, V. Humans and Automation: Use, Misuse, Disuse, Abuse. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 39, 2 (June 1997), 230–253.
- [237] PARASURAMAN, R., SHERIDAN, T., AND WICKENS, C. A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 30, 3 (May 2000), 286–297. Conference Name: IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans.
- [238] PARK, T. M., WON, S. Y., LEE, S. R., AND SZIEBIG, G. Force feedback based gripper control on a robotic arm. In *2016 IEEE 20th Jubilee International Conference on Intelligent Engineering Systems (INES)* (June 2016), pp. 107–112.
- [239] PAYNE, C. J., LATT, W. T., AND YANG, G.-Z. A new hand-held force-amplifying device for micromanipulation. In *2012 IEEE International Conference on Robotics and Automation* (May 2012), pp. 1583–1588. ISSN: 1050-4729.

- [240] PAYNE, C. J., VYAS, K., BAUTISTA-SALINAS, D., ZHANG, D., MARCUS, H. J., AND YANG, G.-Z. Shared-Control Robots. In *Neurosurgical Robotics*, H. J. Marcus and C. J. Payne, Eds., Neuromethods. Springer US, New York, NY, 2021, pp. 63–79.
- [241] PELLETIER, C., AND OLIVER, M. Learning to play in digital games. *Learning, Media and Technology* 31, 4 (Dec. 2006), 329–342.
- [242] PERROW, C. *Normal Accidents*. Princeton University Press, Oct. 2011.
- [243] PFLEGING, B., RANG, M., AND BROY, N. Investigating user needs for non-driving-related activities during automated driving. In *Proceedings of the 15th International Conference on Mobile and Ubiquitous Multimedia* (New York, NY, USA, Dec. 2016), MUM ’16, Association for Computing Machinery, pp. 91–99.
- [244] PILGRIM, W. *Alienated*. Game [Windows], 2008. OneSwitch.org.uk, Essex, UK.
- [245] PLATINUMGAMES. *Bayonetta*. Game [Wii U], September 2014. PlatinumGames, Osaka, Japan.
- [246] PORTER, J. R., AND KIENTZ, J. A. An empirical study of issues and barriers to mainstream video game accessibility. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility* (New York, NY, USA, Oct. 2013), ASSETS ’13, Association for Computing Machinery, pp. 1–8.

- [247] PURPLE JAM. *Audio Pong*. Game [Mobile], June 2020. Purple Jam, Nottingham, UK.
- [248] RAKHIMKUL, S., KIM, A., PAZYLBEKOV, A., AND SHINTEMIROV, A. Autonomous Object Detection and Grasping Using Deep Learning for Design of an Intelligent Assistive Robot Manipulation System. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)* (Oct. 2019), pp. 3962–3968. ISSN: 2577-1655.
- [249] RASMUSSEN, J. Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models. *IEEE Transactions on Systems, Man, and Cybernetics SMC-13*, 3 (May 1983), 257–266.
- [250] RAZBUTEN. What Games Are Like For Someone Who Doesn't Play Games, Sept. 2019.
- [251] RICE, M., YAU, L. J., ONG, J., WAN, M., AND NG, J. Intergenerational gameplay: evaluating social interaction between younger and older players. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems* (New York, NY, USA, May 2012), CHI EA '12, Association for Computing Machinery, pp. 2333–2338.
- [252] RICHARDS, C., AND GRAHAM, T. N. Developing Compelling Repetitive-Motion Exergames by Balancing Player Agency with the Constraints of Exercise. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems* (New York, NY, USA, June 2016), DIS '16, Association for Computing Machinery, pp. 911–923.

- [253] RIVERS, A., MOYER, I. E., AND DURAND, F. Position-correcting tools for 2D digital fabrication. *ACM Transactions on Graphics* 31, 4 (July 2012), 88:1–88:7.
- [254] ROBERTS, T. T., LEONARD, G. R., AND CEPELA, D. J. Classifications In Brief: American Spinal Injury Association (ASIA) Impairment Scale. *Clinical Orthopaedics and Related Research* 475, 5 (May 2017), 1499–1504.
- [255] ROOPCHAND-MARTIN, S., NELSON, G., AND GORDON, C. Can persons with paraplegia obtain training heart rates when boxing on the Nintendo Wii?(CASE STUDY)(Author abstract). *New Zealand Journal of Physiotherapy* 42, 1 (Mar. 2014), 28–32.
- [256] ROSENQVIST, R., BOLDSEN, J., PAPACHRISTOS, E., AND MERRITT, T. Mete orQuest - Bringing Families Together Through Proxemics Play In A Mobile Social Game. In *Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play* (New York, NY, USA, Oct. 2018), CHI PLAY ’18, Association for Computing Machinery, pp. 439–450.
- [257] ROWLAND, J. L., MALONE, L. A., FIDOPIASTIS, C. M., PADALABALA-NARAYANAN, S., THIRUMALAI, M., AND RIMMER, J. H. Perspectives on Active Video Gaming as a New Frontier in Accessible Physical Activity for Youth With Physical Disabilities. *Physical Therapy* 96, 4 (Apr. 2016), 521–532.
- [258] ROWLAND, J. L., AND RIMMER, J. H. Feasibility of Using Active Video Gaming as a Means for Increasing Energy Expenditure in Three Nonambulatory Young Adults With Disabilities. *PM&R* 4, 8 (2012), 569–573.

- [259] ROY, B., GRAHAM, N., AND GUTWIN, C. DiscoTech: a plug-in toolkit to improve handling of disconnection and reconnection in real-time groupware. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work* (New York, NY, USA, Feb. 2012), CSCW '12, Association for Computing Machinery, pp. 1287–1296.
- [260] RUSSELL, S. *Spacewar!* Game [PDP-1], February 1962. Steve Russell, Cambridge, Massachusetts, USA.
- [261] SALVADOR, T., SCHOLTZ, J., AND LARSON, J. The Denver model for groupware design. *ACM SIGCHI Bulletin* 28, 1 (Jan. 1996), 52–58.
- [262] SANDERS ASSOCIATES. *Magnavox Odyssey*. Game [Console], September 1972. Magnavox, Knoxville, Tennessee, USA.
- [263] SARTER, N. B., AND WOODS, D. D. Pilot Interaction With Cockpit Automation II: An Experimental Study of Pilots' Model and Awareness of the Flight Management System. *The International Journal of Aviation Psychology* 4, 1 (Jan. 1994), 1–28. Publisher: Taylor & Francis eprint: [https://doi.org/10.1207/s15327108ijap0401\\_1](https://doi.org/10.1207/s15327108ijap0401_1).
- [264] SARTER, N. B., AND WOODS, D. D. How in the World Did We Ever Get into That Mode? Mode Error and Awareness in Supervisory Control. *Human Factors* 37, 1 (Mar. 1995), 5–19. Publisher: SAGE Publications Inc.
- [265] SARTER, N. B., AND WOODS, D. D. Team Play with a Powerful and Independent Agent: Operational Experiences and Automation Surprises on the

- Airbus A-320. *Human Factors* 39, 4 (Dec. 1997), 553–569. Publisher: SAGE Publications Inc.
- [266] SARTER, N. B., WOODS, D. D., AND BILLINGS, C. E. Automation surprises. *Handbook of human factors and ergonomics* 2 (1997), 1926–1943.
- [267] SAYENKO, D., MASANI, K., MILOSEVIC, M., ROBINSON, M., VETTE, A., MC CONVILLE, K., AND POPOVIC, M. Video game-based neuromuscular electrical stimulation system for calf muscle training: A case study. *Medical Engineering & Physics* 33, 2 (Mar. 2011), 249–255.
- [268] SCHNEIDER, A. L. J., KEIVER, K., PRITCHARD ORR, A., REYNOLDS, J. N., GOLUBOVICH, N., AND GRAHAM, T. N. Toward the Design of Enjoyable Games for Children with Fetal Alcohol Spectrum Disorder. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA, Apr. 2020), CHI ’20, Association for Computing Machinery, pp. 1–13.
- [269] SCHULMAN, J., WOLSKI, F., DHARIWAL, P., RADFORD, A., AND KLIMOV, O. Proximal policy optimization algorithms, 2017.
- [270] SEARS, A., AND JACKO, J. A. *Human-computer interaction*. CRC Press, Boca Raton, FL, 2009. OCLC: 1330994974.
- [271] SENO, T., OKUOKA, K., OSAWA, M., AND IMAI, M. Adaptive Semi-autonomous Agents via Episodic Control. In *Proceedings of the 6th International Conference on Human-Agent Interaction* (New York, NY, USA, Dec. 2018), HAI ’18, Association for Computing Machinery, pp. 377–379.

- [272] SENTOUH, C., NGUYEN, A.-T., FLORIS, J., AND POPIEUL, J.-C. Multiple Controller Switching Concept for Human-Machine Shared Control of Lane Keeping Assist Systems. In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (Oct. 2018), pp. 2730–2735. ISSN: 2577-1655.
- [273] SHERIDAN, T. B. *Telerobotics, automation, and human supervisory control*. MIT Press, Cambridge, Mass, 1992.
- [274] SHERIDAN, T. B., AND VERPLANK, W. L. Human and Computer Control of Undersea Teleoperators. Tech. rep., MASSACHUSETTS INST OF TECH CAMBRIDGE MAN-MACHINE SYSTEMS LAB, July 1978. Section: Technical Reports.
- [275] SHILKROT, R., MAES, P., PARADISO, J. A., AND ZORAN, A. Augmented Airbrush for Computer Aided Painting (CAP). *ACM Transactions on Graphics* 34, 2 (Mar. 2015), 19:1–19:11.
- [276] SIYAHHAN, S., BARAB, S. A., AND DOWNTON, M. P. Using activity theory to understand intergenerational play: The case of Family Quest. *International Journal of Computer-Supported Collaborative Learning* 5, 4 (Dec. 2010), 415–432.
- [277] SMITH, G., WHITEHEAD, J., AND MATEAS, M. Tanagra: a mixed-initiative level design tool. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games* (New York, NY, USA, June 2010), FDG '10, Association for Computing Machinery, pp. 209–216.

- [278] SMITH, K. *Digital Outcasts*. Elsevier Science & Technology, San Francisco, 2013. OCLC: 852527321.
- [279] SOH, H., AND DEMIRIS, Y. Learning assistance by demonstration: smart mobility with shared control and paired haptic controllers. *Journal of Human-Robot Interaction* 4, 3 (Dec. 2015), 76–100.
- [280] SPIELMAN, Z., AND LE BLANC, K. Boeing 737 MAX: Expectation of Human Capability in Highly Automated Systems. In *Advances in Human Factors in Robots, Drones and Unmanned Systems* (Cham, 2021), M. Zallio, Ed., Advances in Intelligent Systems and Computing, Springer International Publishing, pp. 64–70.
- [281] STACH, T., GUTWIN, C., PINELLE, D., AND IRANI, P. Improving Recognition and Characterization in Groupware with Rich Embodiments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2007), CHI ’07, ACM, pp. 11–20. event-place: San Jose, California, USA.
- [282] STEELE, M., AND GILLESPIE, R. B. Shared Control between Human and Machine: Using a Haptic Steering Wheel to Aid in Land Vehicle Guidance. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 45, 23 (Oct. 2001), 1671–1675. Publisher: SAGE Publications Inc.
- [283] STEVENS, G., BOSSAUER, P., VONHOLDT, S., AND PAKUSCH, C. Using Time and Space Efficiently in Driverless Cars. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (2019), pp. 1–14.

- [284] STRAUCH, B. Ironies of Automation: Still Unresolved After All These Years. *IEEE Transactions on Human-Machine Systems* 48, 5 (Oct. 2018), 419–433.
- [285] STUART MORRIS, GAIL FAWCETT, LAURENT BRISEBOIS, AND JEFFREY HUGHES. A demographic, employment and income profile of Canadians with disabilities aged 15 years and over, 2017. Tech. rep., Statistics Canada, Nov. 2018.
- [286] SWITKES, J. P., ROSSETTER, E. J., COE, I. A., AND GERDES, J. C. Handwheel Force Feedback for Lanekeeping Assistance: Combined Dynamics and Stability. *Journal of Dynamic Systems, Measurement, and Control* 128, 3 (Sept. 2006), 532–542.
- [287] TACCOLA, G., SAYENKO, D., GAD, P., GERASIMENKO, Y., AND EDGERTON, V. R. And yet it moves: Recovery of volitional control after spinal cord injury. *Progress in Neurobiology* 160 (Jan. 2018), 64–81.
- [288] TAHBOUB, K. Natural and manmade shared-control systems: an overview. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)* (May 2001), vol. 3, pp. 2655–2660 vol.3. ISSN: 1050-4729.
- [289] TAITO. *Space Invaders*. Game [Arcade], April 1978. Taito, Tokyo, Japan.
- [290] TAMAKI, E., MIYAKI, T., AND REKIMOTO, J. PossessedHand: techniques for controlling human hands using electrical muscles stimuli. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New

- York, NY, USA, May 2011), CHI '11, Association for Computing Machinery, pp. 543–552.
- [291] TAYLOR, R., JENSEN, P., WHITCOMB, L., BARNES, A., KUMAR, R., STOIANOVICI, D., GUPTA, P., WANG, Z., DEJUAN, E., AND KAVOUSSI, L. A Steady-Hand Robotic System for Microsurgical Augmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI'99* (Berlin, Heidelberg, 1999), C. Taylor and A. Colchester, Eds., Lecture Notes in Computer Science, Springer, pp. 1031–1041.
- [292] TEKİNBAŞ, K. S., AND ZIMMERMAN, E. *Rules of play: game design fundamentals*. MIT Press, Cambridge, Mass, 2003.
- [293] TENG, S.-Y., LIANG, R.-H., TANG, J.-H., CHEN, B.-Y., AND LIAO, Y.-C. Playing Air Guitar by Electrical Muscle Stimulation. *AUI 2016: 1 st Asian Workshop on User Interface*, 2.
- [294] TESLA. Introducing Navigate on Autopilot, Oct. 2018.
- [295] THE ABLEGAMERS FOUNDATION INC. Accessible player experiences, 2022.
- [296] THIRUMALAI, M., KIRKLAND, W. B., MISKO, S. R., PADALABALANARAYANAN, S., AND MALONE, L. A. Adapting the Wii Fit Balance Board to Enable Active Video Game Play by Wheelchair Users: User-Centered Design and Usability Evaluation. *JMIR Rehabilitation and Assistive Technologies* 5, 1 (Mar. 2018), e2.
- [297] TIEL. HAPTIC INTELLIGENTSIA – TIEL, 2021.

- [298] TOUPS, Z. O., HAMMER, J., HAMILTON, W. A., JARRAH, A., GRAVES, W., AND GARRETSON, O. A framework for cooperative communication game mechanics from grounded theory. In *Proceedings of the first ACM SIGCHI annual symposium on Computer-human interaction in play* (New York, NY, USA, Oct. 2014), CHI PLAY '14, Association for Computing Machinery, pp. 257–266.
- [299] TRAUTMAN, P. Assistive Planning in Complex, Dynamic Environments: A Probabilistic Approach. In *2015 IEEE International Conference on Systems, Man, and Cybernetics* (Oct. 2015), pp. 3072–3078.
- [300] TUNSTEL, E., MAIMONE, M., TREBI-OLLENNU, A., YEN, J., PETRAS, R., AND WILLSON, R. Mars Exploration Rover mobility and robotic arm operational performance. In *2005 IEEE International Conference on Systems, Man and Cybernetics* (Oct. 2005), vol. 2, pp. 1807–1814 Vol. 2. ISSN: 1062-922X.
- [301] UNITED STATES. PRESIDENT'S COMMISSION ON THE ACCIDENT AT THREE MILE ISLAND. *Report of the President's Commission on the Accident at Three Mile Island : the need for change : the legacy of TMI*. Washington : President's Commission on the Accident at Three Mile Island, 1979., 1979.
- [302] URBANEK, M., AND GÜLDENPFENNIG, F. Unpacking the Audio Game Experience: Lessons Learned from Game Veterans. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play* (New York, NY, USA, Oct. 2019), CHI PLAY '19, Association for Computing Machinery, pp. 253–264.
- [303] VAN DIGGELEN, J., AND JOHNSON, M. Team Design Patterns. In *Proceedings of the 7th International Conference on Human-Agent Interaction* (New York, NY,

- USA, Sept. 2019), HAI '19, Association for Computing Machinery, pp. 118–126.
- [304] VANDEN ABEELE, V., AND DE SCHUTTER, B. Designing intergenerational play via enactive interaction, competition and acceleration. *Personal and Ubiquitous Computing* 14, 5 (July 2010), 425–433.
- [305] VANSICKLE, D., COOPER, R., AND STER, J. Wheelchair virtual joystick interface. In *Proceedings of 17th International Conference of the Engineering in Medicine and Biology Society* (Montreal, Que., Canada, 1995), vol. 2, IEEE, pp. 1175–1176.
- [306] VICENCIO-MOREIRA, R., MANDRYK, R. L., GUTWIN, C., AND BATEMAN, S. The effectiveness (or lack thereof) of aim-assist techniques in first-person shooter games. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2014), CHI '14, Association for Computing Machinery, p. 937–946.
- [307] VILLAREALE, J., HARTEVELD, C., AND ZHU, J. "I Want To See How Smart This AI Really Is": Player Mental Model Development of an Adversarial AI Player. *Proceedings of the ACM on Human-Computer Interaction* 6, CHI PLAY (Oct. 2022), 219:1–219:26.
- [308] VOIDA, A., AND GREENBERG, S. Wii all play: the console game as a computational meeting place. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2009), CHI '09, Association for Computing Machinery, pp. 1559–1568.

- [309] VOIDA, A., AND GREENBERG, S. Console gaming across generations: exploring intergenerational interactions in collocated console gaming. *Universal Access in the Information Society* 11, 1 (Mar. 2012), 45–56.
- [310] VOLVO. Lane Keeping Aid | Lane Keeping Aid | Driver support | XC60 2021 Early | Volvo Support, 2020.
- [311] WALL, T., FEINN, R., CHUI, K., AND CHENG, M. S. The effects of the Nintendo™ Wii Fit on gait, balance, and quality of life in individuals with incomplete spinal cord injury. *The Journal of Spinal Cord Medicine* 38, 6 (Nov. 2015), 777–783.
- [312] WALLER, S., BRADLEY, M., HOSKING, I., AND CLARKSON, P. J. Making the case for inclusive design. *Applied Ergonomics* 46 (Jan. 2015), 297–303.
- [313] WANG, B., TAYLOR, L., AND SUN, Q. Families that play together stay together: Investigating family bonding through video games. *New Media & Society* 20, 11 (Nov. 2018), 4074–4094. Publisher: SAGE Publications.
- [314] WANG, H., AND LIU, X. P. Adaptive Shared Control for a Novel Mobile Assistive Robot. *IEEE/ASME Transactions on Mechatronics* 19, 6 (Dec. 2014), 1725–1736. Conference Name: IEEE/ASME Transactions on Mechatronics.
- [315] WARHORSE STUDIOS. *Kingdom Come: Deliverance*. Game [PC], February 2018. Warhorse Studios, Prague, Czech Republic.
- [316] WATSON, N. Procedural Elaboration: How Players Decode Minecraft. *Loading...* 10, 16 (Feb. 2017). Number: 16.

- [317] WHALLEY HAMMELL, K. Experience of rehabilitation following spinal cord injury: a meta-synthesis of qualitative findings. *Spinal Cord* 45, 4 (Apr. 2007), 260–274. Number: 4 Publisher: Nature Publishing Group.
- [318] WIEGAND, G., HOLLÄNDER, K., RUPP, K., AND HUSSMANN, H. The Joy of Collaborating with Highly Automated Vehicles. In *12th International Conference on Automotive User Interfaces and Interactive Vehicular Applications* (New York, NY, USA, Sept. 2020), AutomotiveUI '20, Association for Computing Machinery, pp. 223–232.
- [319] WIENER, E. L., UNITED STATES, NATIONAL AERONAUTICS AND SPACE ADMINISTRATION, UNITED STATES, NATIONAL TECHNICAL INFORMATION SERVICE, AND AMES RESEARCH CENTER. *Human factors of advanced technology ("Glass cockpit") transport aircraft*. NASA Ames research Center ; For sale by the National Technical Information Service, Washington, DC; Springfield, VA, 1989. OCLC: 224707113.
- [320] WINTERHOLLER, J., JANNA, B., PETROV, A., AND MAIER, T. Innovative Haptic Interface for Navigation. Accepted: 2014-06-26T23:07:27Z Publisher: California State University, Northridge.
- [321] WINTERSBERGER, P., FRISON, A.-K., RIENER, A., AND SAWITZKY, T. v. Fostering User Acceptance and Trust in Fully Automated Vehicles: Evaluating the Potential of Augmented Reality. *Presence* 27, 1 (Mar. 2019), 46–62. Conference Name: Presence.
- [322] WISANUVEJ, P., GIATAGANAS, P., LEIBRANDT, K., LIU, J., HUGHES, M., AND YANG, G.-Z. Three-dimensional robotic-assisted endomicroscopy with a

- force adaptive robotic arm. In *2017 IEEE International Conference on Robotics and Automation (ICRA)* (May 2017), pp. 2379–2384.
- [323] WOBBROCK, J. O., GAJOS, K. Z., KANE, S. K., AND VANDERHEIDEN, G. C. Ability-based design. *Communications of the ACM* 61, 6 (May 2018), 62–71.
- [324] WOBBROCK, J. O., KANE, S. K., GAJOS, K. Z., HARADA, S., AND FROEHLICH, J. Ability-Based Design: Concept, Principles and Examples. *ACM Trans. Access. Comput.* 3, 3 (Apr. 2011), 9:1–9:27.
- [325] WOODS, J., AND WALTON, D. Chapter 9. Post Hoc, Ergo Propter Hoc. In *Chapter 9. Post Hoc, Ergo Propter Hoc*. De Gruyter Mouton, Nov. 2019, pp. 121–142.
- [326] WUERTZ, J., ALHARTHI, S. A., HAMILTON, W. A., BATEMAN, S., GUTWIN, C., TANG, A., TOUPS, Z., AND HAMMER, J. A Design Framework for Awareness Cues in Distributed Multiplayer Games. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2018), CHI ’18, Association for Computing Machinery, pp. 1–14.
- [327] YAMAOKA, J., AND KAKEHI, Y. dePENd: augmented handwriting system using ferromagnetism of a ballpoint pen. In *Proceedings of the 26th annual ACM symposium on User interface software and technology* (New York, NY, USA, Oct. 2013), UIST ’13, Association for Computing Machinery, pp. 203–210.
- [328] YANNAKAKIS, G. N., AND TOGELIUS, J. *Artificial Intelligence and Games*. Springer International Publishing, 2018.

- [329] YUAN, B., FOLMER, E., AND HARRIS, F. C. Game accessibility: a survey. *Universal Access in the Information Society* 10, 1 (June 2010), 81–100.
- [330] YUAN, B., FOLMER, E., AND HARRIS, F. C. Game accessibility: a survey. *Universal Access in the Information Society* 10, 1 (Mar. 2011), 81–100.
- [331] ZORAN, A., AND PARADISO, J. A. FreeD: a freehand digital sculpting tool. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, Apr. 2013), CHI ’13, Association for Computing Machinery, pp. 2613–2616.

---

## Appendices

---

## A Health Sciences Research Ethics Board (HSREB) Approval

**Queen's University Health Sciences & Affiliated Teaching Hospitals Research Ethics Board (HSREB)****HSREB Initial Ethics Clearance**

July 15, 2019

Mr. Gabriele Cimolino  
School of Computing  
Queen's University

**TRAQ #:** 6027049

**Department Code:** SCOMP-004-19

**Study Title:** "SCOMP-004-19 INPUT AUTOMATION AS GAMEPLAY ASSISTANCE FOR PEOPLE WITH SPINAL CORD INJURY"

**Supervisor:** Dr. T C Nicholas Graham

**Co-Investigators:** Dr. Sussan Askari

**Review Type:** Delegated

**Date Ethics Clearance Issued:** July 15, 2019

**Ethics Clearance Expiry Date:** July 15, 2020

Dear Mr. Cimolino:

The Queen's University Health Sciences & Affiliated Teaching Hospitals Research Ethics Board (HSREB) has reviewed the application and granted ethics clearance for the documents listed below. Ethics clearance is granted until the expiration date noted above.

- Protocol: v.2019JUL13
- Letter of Information: v.2019JUL13
- Consent Form v.2019JUL13
- Gameplay Experience Questionnaire - PXI Dynamics v.2019JUN24
- Gameplay Experience Questionnaire - PXI Aesthetics v.2019JUN24
- Demographic Questionnaire: v.2019JUL13
- Interview Questions v.2019JUL24

**Documents Acknowledged:**

- Ethics Training/CORE Certificates: Graham, Askari, Cimolino
- Confidentiality Agreement v.2019JUN24
- MOTOMed viva2 Device Manual

**Amendments:** No deviation from, or changes to the protocol, informed consent form and conduct of study should be initiated without prior written clearance or an appropriate amendment from the HSREB, except when necessary to eliminate immediate hazard(s) to study participants or when the change(s) involves only administrative or logistical aspects of the study.

**Renewals:** An annual renewal event form or a study closure event form must be submitted annually as per the TCPS

2 2014 Article 6.14. As a courtesy, the Office of Research Ethics may send reminders 30 days in advance of the ethics clearance expiry date. All lapses in ethics clearance will be documented on the annual renewal clearance letter. Suspension letters may be issued for lapses in ethics clearances one day or greater, with subsequent termination and closure of the ethics file for lapses greater than 10 business days. Terminations should be reported to applicable regulatory authorities (e.g., Health Canada, FDA).

**Completion/Termination:** The HSREB must be notified of the completion or termination of this study through the submission of a study closure event in TRAQ.

**Reporting of Serious Adverse Events:** Any unexpected serious adverse events occurring locally must be reported within 2 working days or earlier if required by the study sponsor. All other serious adverse events must be reported within 15 days after becoming aware of the information.

**Reporting of Complaints:** Any complaints made by participants or persons acting on behalf of participants must be reported to the Research Ethics Board within 7 days of becoming aware of the complaint.

**Note: All documents supplied to participants must have the contact information for the Research Ethics Board.**

Investigators please note that if your study is registered by the sponsor, you must take responsibility to ensure that the registration information is accurate and complete.

Regards,



Albert F. Clark, PhD  
Chair, Queen's University Health Sciences and Affiliated Teaching Hospitals Ethics Board

The HSREB operates in compliance with, and is constituted in accordance with, the requirements of the Tri-Council Policy Statement: Ethical Conduct for Research Involving Humans (TCPS 2 2014); the international Conference on Harmonisation Good Clinical Practice Consolidated Guideline (ICH GCP); Part C, Division 5 of the Food and Drug Regulations; Part 4 of the Natural Health Product Regulations; Part 3 of the Medical Devices Regulations, and the provisions of the Ontario Personal Health Information Protection Act (PHIPA 2004) and its applicable regulations. The HSREB is qualified through the CTO REB Qualification Program and is registered with the U.S. Department of Health and Human Services (DHHS) Office for Human Research Protection (OHRP). Federalwide Assurance Number: FWA#: 00004184, IRB#: 00001173. HSREB members involved in the research project do not participate in the review, discussion or decision.

**B General Research Ethics Board (GREB) Approval**



December 21, 2021

Mr. Gabriele Cimolino  
Queen's University

**Title: "GCOMP-116-21 Video games for people who don't play video games;" TRAQ # 6035229**

Dear Mr. Cimolino:

The General Research Ethics Board (GREB), by means of a delegated board review, has cleared your proposal entitled "**GCOMP-116-21 Video games for people who don't play video games**" for ethical compliance with the Tri-Council Guidelines (TCPS 2) and Queen's ethics policies. In accordance with the Tri-Council Guidelines (Article 6.14) and Standard Operating Procedures (405), your project has been cleared for one year.

You are reminded of your obligation to submit an annual renewal form prior to the annual renewal due date (access this form at <http://www.queensu.ca/traq/signon.html>; click on "Events;" under "Create New Event" click on "General Research Ethics Board Annual Renewal/Closure Form for Cleared Studies"). Please note that when your research project is completed, you need to submit an Annual Renewal/Closure Form in Romeo/traq indicating that the project is 'completed' so that the file can be closed. This should be submitted at the time of completion; there is no need to wait until the annual renewal due date.

You are reminded of your obligation to advise the GREB of any adverse event(s) that occur during this one-year period (access this form at <http://www.queensu.ca/traq/signon.html>; click on "Events;" under "Create New Event" click on "General Research Ethics Board Adverse Event Form"). An adverse event includes, but is not limited to, a complaint, a change or unexpected event that alters the level of risk for the researcher or participants or situation that requires a substantial change in approach to a participant(s). You are also advised that all adverse events must be reported to the GREB within 48 hours.

You are also reminded that all changes that might affect human participants must be cleared by the GREB. For example, you must report changes to the level of risk, applicant characteristics, and implementation of new procedures. To submit an amendment form, access the application by at <http://www.queensu.ca/traq/signon.html>; click on "Events;" under "Create New Event" click on "General Research Ethics Board Request for the Amendment of Approved Studies." Once submitted, these changes will automatically be sent to the Ethics Coordinator, GREB, at University Research Services for further review and clearance by GREB or the Chair, GREB.

On behalf of the General Research Ethics Board, I wish you continued success in your research.

Sincerely,

A handwritten signature in blue ink, appearing to read "Dean A. Tripp".

Professor Dean A. Tripp, PhD  
Chair, General Research Ethics Board (GREB)  
Departments of Psychology, Anesthesiology & Urology  
Queen's University