Shared Control for Game Accessibility: Understanding Current Human Cooperation Practices to Inform the Design of Partial Automation Solutions

Dragan Ahmetovic
Dept. of Computer Science
Università degli studi di Milano
Milano, Italy
dragan.ahmetovic@unimi.it

Matteo Manzoni
Dept. of Computer Science
Università degli Studi di Milano
Milano, Italy
matteo.manzoni@unimi.it

Filippo Corti
Dept. of Computer Science
Università degli Studi di Milano
Milan, Italy
filippo.corti1@studenti.unimi.it

Sergio Mascetti
Dept. of Computer Science
Università degli Studi di Milano
Milano, Italy
sergio.mascetti@unimi.it

Abstract

Shared control is a form of video gaming accessibility support that allows players with disabilities to delegate inaccessible controls to another person. Through interviews involving 14 individuals with lived experience of accessible gaming in shared control, we explore the ways in which shared control technologies are adopted in practice, the accessibility challenges they address, and how the support currently provided in shared control can be automated to remove the need for a human assistant. Findings indicate that shared control is essential for enabling access to otherwise inaccessible games, but its reliance on human support is a key limitation. Participants welcomed the idea of automating the support with software agents, while also identifying limitations and design requirements. Accordingly, this work contributes insights into current practices and proposes guidelines for developing automated support systems.

CCS Concepts

• **Applied computing** → *Computer games*; • **Human-centered computing** → *Accessibility technologies*; **Collaborative interaction**; • **Social and professional topics** → *People with disabilities*.

Keywords

Shared control; partial automation; human cooperation; accessible gaming.

ACM Reference Format:
Dragan Ahmetovic, Matteo Manzoni, Filippo Corti, and Sergio Mascetti. 2026. Shared Control for Game Accessibility: Understanding Current Human Cooperation Practices to Inform the Design of Partial Automation Solutions. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26), April 13–17, 2026, Barcelona, Spain.* ACM, New York, NY, USA, 19 pages. https://doi.org/10.1145/3772318.3791080

CHI '26, Barcelona, Spain

ACM ISBN 979-8-4007-2278-3/26/04
https://doi.org/10.1145/3772318.3791080

1 Introduction

Video games are a leading entertainment industry, with more than 3.22 billion players worldwide [77], 427 million of whom had some form of disability in 2023 [61]. Numerous assistive technologies have been developed to enable autonomous access to video games by players with disabilities [6]. Despite these technologies, not all players are able to access every game independently [13, 58].

To address this issue, *Shared Control* solutions, such as *Xbox Controller Assist* [60], allow players with disabilities to delegate control of some game actions to another person. Recent research also explores how to replace human support with a software agent [19]. However, to the best of our knowledge, no prior study investigates how existing shared control technologies are commonly used by players, to address which accessibility challenges, and how these needs can be translated into technologies that automate such support.

Consequently, we advance the research by conducting interviews and focus groups with 14 individuals who have prior experience in the use of shared control systems, including people with disabilities who play using shared control, others who support players with disabilities during shared control gaming, and experts in assistive gaming technologies. Through reflexive thematic analysis, the research first aims to gain a deeper understanding of the shared control technologies currently in use, highlighting their benefits and limitations. Second, we investigate the acceptability and the key features that a system automating the role of the supporting person should have.

Results show that shared control is essential for making certain games accessible that would otherwise not be playable. However, these techniques have a major limitation: they require the availability of a person willing to provide support. For this reason, participants welcomed the possibility of replacing the supporting person with a software agent, while at the same time highlighting both the potential limitations of the solution and the features it should possess. Our work, therefore, provides a better understanding of the current use of shared control systems and outlines design guidelines for the development of shared control technologies with automated support.

In summary, the key contributions introduced in this paper are:

- Formative interviews and focus groups with 14 participants, including people with disabilities, their caregivers, and accessibility experts, on their experience with using Shared Control as a means to achieve accessible gaming.
- Results of a reflexive thematic analysis conducted on the transcripts of the interviews, identifying key themes related to the use of Shared Control for gaming, and the possibility of automating shared control support.
- Design recommendations to guide the development of new Shared Control solutions providing human support as well as automated assistance.

2 Background

This section surveys the accessibility problems faced by gamers with disabilities in playing video games (Section 2.1) and the existing assistive technologies designed to overcome these limitations (Section 2.2), with a specific focus on shared control solutions (Section 2.3) Additionally, it examines the trade-off between accessibility and autonomy when people with disabilities are supported by a human assistant (Section 2.4).

2.1 Video Games Accessibility

Playing video games poses accessibility challenges for people with disabilities [6, 12, 15, 36, 58, 98]. These challenges include single-sense feedback (*e.g.*, audio cues without visual counterparts) [12, 15, 36, 58], unsuitable button layouts [15, 25, 58], and demanding input gestures, such as repeated button mashing or long holds [12, 15]. Thus, gamers with disabilities have to adapt each game to their specific needs [58]. In some cases, however, adaptations fall short, forcing players to "play their own game", engaging it in unconventional ways [36], or even to abandon the game altogether [58]. Guidelines written to facilitate the development of in-game accessibility features exist [9, 12, 13, 36, 58, 63, 84, 93] but are rarely implemented [6, 98]. Due to this, accessibility often needs to be achieved through outside-of-game solutions, like accessible game controllers and input-modifying software.

2.2 Accessibility of Video Game Controls

Various peripherals are used as video game controllers. In particular, gamepads provided with major gaming consoles have evolved to share several common design features [54, 56]. For example, Xbox and PlayStation controllers are designed to be held with both hands and require coordinated use of the thumbs (each capable of controlling a joystick and four buttons), as well as two additional fingers (usually the index and middle fingers) to operate two side buttons per hand. This design assumes that all users have similar hand anatomy and size [16], can operate multiple inputs with both hands [25], and are able to perform quick, coordinated, and reactive movements [25]. As a result, standard controllers may be inaccessible to players with disabilities, for example, those with upper-limb mobility impairments.

Alternative controllers, designed to be more accessible and customizable, offer different button layouts from standard ones. Some are designed to be operated with a single hand [65] or placed on a flat surface [1, 31, 64]. Others are personalizable through external buttons [24, 31, 64] or feature a modular design [17]. Software support has also been proposed to enhance the accessibility of game controllers. Some tools allow remapping controller buttons to different commands; these are sometimes available directly within the games themselves. Others allow remapping of commands across different types of input devices. For example, JoyToKey [45] allows a game controller to emulate keyboard inputs for games designed to be played with a keyboard. Further solutions enable users to perform actions using voice input [7, 95] or facial expressions [57, 74], which can then be mapped to specific game commands.

Using these tools, inaccessible games can be made accessible to many people with disabilities. The resulting game setup – that is, the specific hardware and software configuration of accommodations and accessibility tools used – can be quite complex and can vary based on the game, player's disability, and preferences. However, for some gamers with disabilities, existing accessibility tools may not suffice to effectively access all the inputs required to play a game [19]. In such cases, one possibility is to delegate inaccessible game controls to someone else through shared control [21, 60].

2.3 Shared Control

Shared control refers to two or more agents collaboratively interacting with a system. Beyond video games, shared control has been studied for collaborative robot control [10, 29, 39], assisted vehicle driving [51], control of mobility aids such as wheelchairs [49] or guide robots [47], surgical augmentation [88], and management of shared user interfaces [27]. In these contexts, the proposed interaction models involve either cooperation between humans or the integration of software agents to assist the user. The terminology used to distinguish between different forms of support varies across application domains. For example, when referring to cooperation between only human actors, commonly used terms include *human-human collaboration* [97], *multi-user interaction* [27], and *multi-operator-single-robot* [33]. Similarly, systems in which software agents contribute to the control are referred to using terms such as *partial automation* [21], *human-AI collaboration* [97], *human-in-the-loop* [39], or *supervisory control* [10]. To disambiguate the various meanings, we use the term *shared control* to refer to all interactions that allow game controls to be distributed across multiple actors, usually a *pilot*, *i.e.*, the person primarily responsible for playing (typically a person with disabilities), and a *copilot* supporting the pilot. As illustrated in Figure 1, the two subtypes of shared control are: *human cooperation*, when the copilot is a human actor; and *partial automation*, when the copilot is a software agent.

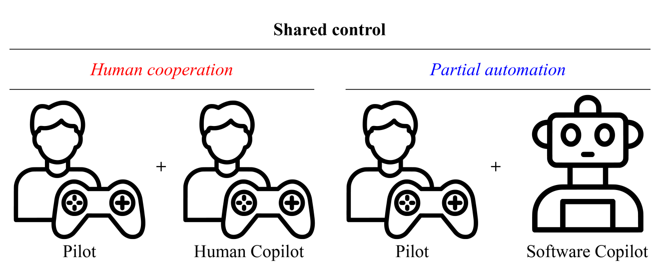

Figure 1: Relationship between shared control, human cooperation, and partial automation

2.3.1 *Human Cooperation for Video Games Accessibility.*

Several human cooperation solutions have been proposed in the context of video games, although not explicitly as assistive technologies. Instead, most studies have focused on analyzing how different forms of cooperation affect social interaction, gameplay experience, and perceived enjoyment. In particular, prior works have observed that introducing interdependence mechanics between players led to higher levels of social interaction compared to independent play [53, 80, 87]. However, requiring cooperation to progress through the game was also found to reduce players' perceived autonomy and control [80].

In the context of video game accessibility, Microsoft has released *Xbox Controller Assist* (formerly *Xbox Copilot*) [60], a software available on Xbox consoles and Windows PCs that allows linking two controllers so that two players can use them to control the same game actions, as if they were using a single controller. This solution is actively advertised as an accessibility feature "to help a friend or loved one through a game or console experience", and evidence of the usage of this solution by players with disabilities is available on various social media[1]. Similarly, on *PlayStation 5* (PS5) consoles, the *PS5 Access Controller* [31] can be paired with another PS5 controller, enabling human cooperation. A third-party solution, the *Titan Two* adapter [23], also supports remote human cooperation over the internet.

Gonçalves et al. [37] have investigated human cooperation related to video game accessibility. Specifically, they developed two games intended to be played in pairs by individuals with mixed abilities: one sighted player would face a visual challenge, while one blind player would face an auditory one. Their findings show that cooperation between players with differing abilities can be effectively supported through game design that leverages each participant's strengths. However, the authors themselves acknowledge that this form of cooperation does not constitute a generalizable accessibility solution, as it relies on custom-designed games and does not apply to disabilities beyond visual impairments.

2.3.2 *Partial Automation for Video Games Accessibility.*

Partial automation has been implemented in commercial video games [4, 30, 72, 73, 86, 92], but it has been mostly limited to specific game tasks. One use case is player balancing: providing assistance that is inversely proportional to the player's skill level, thus ensuring a similar gaming experience for all. For example, some first-person shooters such as *Call of Duty* [4] integrate aim-assistance, whose effectiveness has been evaluated in prior research [44, 94]. Similarly, some racing games implement steering assistance to help players stay on the road [18, 30]. Another approach is to reduce the effort required to play by automating some inputs. For example, some racing games such as *Forza Motorsport* [92] implement automatic gear shifting. More pronounced implementations of partial automation are found in games like the *Bayonetta* series [72, 73], which feature an automatic mode where the player only needs to focus on attacking while the game autonomously handles movement and enemy targeting. Another example is *Zac - O Esquilo* [59], a one-switch game (controlled with only one button) in which the player decides when to move the avatar, but an algorithm determines the direction of the movement. These solutions are specifically designed and implemented for each game, but lack generalizability, which is a significant barrier to the widespread adoption of partial automation as a solution for video game accessibility.

Partial automation has also been explored in previous accessibility research [18–20, 42, 44, 94]. Cimolino et al. [19] employ partial automation in two research video games to demonstrate how distributing gameplay actions between a player and a software agent can enable players with motor disabilities to play. In this approach, the player performs only the actions that they are physically capable of, while the remaining actions are delegated to the software agent. The results of the study highlight that a lack of mutual understanding between the player and the agent can lead to *automation confusion* [20, 70], a phenomenon in which the player struggles to distinguish the outcome of their own actions from those performed by the agent.

These works contribute to a better understanding of how partial automation can support video game accessibility. However, to our knowledge, no previous work explores existing shared control practices, how they are used, and for what purposes. Most importantly, it is unclear how the support currently provided through human cooperation can be provided through generalizable partial automation solutions and how such solutions should be designed.

2.4 Autonomy and Human Assistance for People with Disabilities

Autonomy is a central objective in the design of assistive technologies, even beyond the video games domain. Nevertheless, many people with disabilities require human assistance to accomplish everyday activities such as navigating unfamiliar environments [41], shopping [46, 52], or accessing visual content [11, 14, 32]. Dependence on human helpers can restrict users' autonomy and spontaneity, as activities must often be coordinated with their helpers' availability [82] and expertise [90]. For example, prior work has shown how people with visual impairments may rely on sighted helpers to take photos, introducing coordination challenges [5, 67]. Moreover, receiving assistance from others has social and emotional implications: Shinohara and Wobbrock [82] observed that relying on human support may alter how users are perceived by others and how they perceive themselves, sometimes leading to feelings of burden or reduced agency. For this reason, participants may develop strategies to increase their independence, such as searching online for information about stores before doing in-person shopping [46].

In response to these challenges, recent research has explored how automated assistance can reduce reliance on human helpers while preserving the user's agency [40, 78]. This aligns with arguments that true accessibility is not solely about independence, but also about augmenting users' capabilities to enable meaningful and self-directed engagement [8, 78].

In this paper, we focus on video game accessibility, addressing analogous challenges to those discussed in other domains. Our interviews highlighted similar themes, such as the implications of relying on human support and the potential impact of automating this assistance. For this reason, despite our findings being specific to the video game context, they contribute to the broader discourse on human assistance, automation, and user autonomy for people with disabilities. This is discussed in particular in Section 5.2.1.

[1] A list of social network posts on this topic is available as supplemental material.

Table 1: Participants' Demographic Data. The participant's role is in ID's subscript: P - Pilot, C - Copilot, E - Expert

ID	Age	Gender	Disability		Gameplay		
			Type	Onset Age	Frequency	Difficulties	Platform
P1$_P$	29–38	M	Paresis	Birth	Daily	Moderately	Switch, PC, PS
P2$_P$	29–38	M	Blindness	Birth	Daily	A lot	PC, PS
P3$_P$	29–38	F	Spastic quadriplegia	Birth	Daily	A lot	Xbox
P4$_C$	39–50	M	None	None	Daily	None	Smartphone, Xbox
P5$_E$	39–50	M	Burn injury	20 years	Daily	A little	Xbox
P6$_E$	29–38	M	None	None	Monthly	None	Xbox
P7$_{PC}$	18–28	M	Spastic tetraparesis	Birth	Daily	Moderately	Switch, PS5
P8$_{PC}$	29–38	M	Muscular dystrophy	Birth	Weekly	A lot	PC
P9$_{PC}$	39–50	F	Pediatric tetraparesis	Birth	Daily	Moderately	Smartphone
P10$_{PC}$	18–28	F	Reduced arm mobility	Birth	Daily	A little	Tablet
P11$_{CE}$	39–50	M	None	None	Weekly	None	PC, Xbox
P12$_{CE}$	39–50	M	None	None	Weekly	None	PC
P13$_E$	29–38	F	None	None	Monthly	None	Switch
P14$_E$	39–50	M	None	None	Weekly	None	Switch, Xbox

3 Methodology

To understand how human cooperation technologies are currently used by people with disabilities and to inform the design of future partial automation approaches, semi-structured interviews and focus groups [50] were organized with people who regularly use or have used in the past human cooperation systems for video games. The experimentation was approved by the Ethics Committee of the University of Milan (opinion no. 16/25)[2].

3.1 Participants

Participants were recruited through announcements on social media pages frequented by people with disabilities, for example on *r/disabledgamers* on Reddit, and by reaching out to accessibility specialists' organizations. A local association for people with motor impairments, Spazio Vita[3] of Niguarda Hospital in Milan – Italy, also supported the recruitment by mediating contact with some participants. The following inclusion criteria were adopted:

- Being of legal age.
- Being able to speak Italian or English.
- Having experience with human cooperation in at least one of the following roles:
  - **Pilot**: a player experienced in using human cooperation as an accessibility tool to play video games.
  - **Copilot**: a person experienced in using human cooperation to support a pilot.
  - **Expert**: an accessibility specialist experienced in assisting people with disabilities find accessible game setups for playing video games through human cooperation.

We recruited 14 participants, whose demographic data are reported in Table 1. Ten identified as male, and four as female. The most represented age groups were 29–38 and 39–50 (6 participants each), followed by 18–28 age group (2 participants). Eight participants declared having a disability, among which **P2**$_P$ is the only one with a visual impairment, while the others have motor disabilities.

Participants had various roles (some more than one): 7 were pilots, 5 copilots, and 6 experts. Participant roles are also indicated as subscripts following their IDs: **P** stands for pilot, **C** for copilot, and **E** for expert. For participants with multiple roles, we combined the subscripts (*i.e.*, **CE** for copilot-expert, **PC** for pilot-copilot). In total, 4 individual interviews and 4 focus groups were conducted. Focus groups were conducted in cases when users knew each other and shared usage modalities of human cooperation systems.

- Individual interviews:
  - **P1**$_P$: tried but no longer uses human cooperation, having found a game setup for playing independently.
  - **P2**$_P$: uses remote human cooperation to play with people met online.
  - **P5**$_E$: accessibility specialist with experience in designing new hardware solutions for accessible gaming.
  - **P6**$_E$: assistive technology specialist focusing on adaptive video gaming.
- Focus groups:
  - **P3**$_P$ and **P4**$_C$: **P4**$_C$ is a family member, caregiver, and habitual copilot of **P3**$_P$.
  - **P7**$_{PC}$, **P8**$_{PC}$, **P9**$_{PC}$, and **P10**$_{PC}$: members of a weekly gaming group in which they play cooperatively, acting as pilots and copilots. **P7**$_{PC}$ usually pairs with **P8**$_{PC}$, while **P9**$_{PC}$ plays with **P10**$_{PC}$.
  - **P11**$_{CE}$ and **P12**$_{CE}$: coordinators of the gaming group involving **P7**$_{PC}$, **P8**$_{PC}$, **P9**$_{PC}$ and **P10**$_{PC}$. They sometimes also support the participants as copilots.
  - **P13**$_E$ and **P14**$_E$: work in an association that provides game accessibility support to people with disabilities.

[2]Conducted research is consistent with the general principles enunciated in [3].
[3]https://spaziovitaniguarda.it

All participants play video games at least once a month. Among those with a disability, only **P5**$_E$ and **P10**$_{PC}$ reported having little difficulty in playing video games, while others declared having at least moderate difficulty. The most used platform is Xbox (6 participants), followed by PC (5 participants), Nintendo Switch (4 participants), PlayStation (3 participants), smartphone (2 participants), and tablet (1 participant). For ten people, the reason was the ease of access of the platform (**P1**$_P$, **P2**$_P$, **P3**$_P$, **P4**$_C$, **P6**$_E$, **P8**$_{PC}$, **P9**$_{PC}$, **P10**$_{PC}$, **P11**$_{CE}$, **P14**$_E$). Other motivations include familiarity (**P4**$_C$, **P5**$_E$, **P12**$_{CE}$) and the presence of platform-exclusive video games (**P7**$_{PC}$, **P13**$_E$). Eight (of nine) participants with disabilities also use hardware or software tools to facilitate video game access. These include Xbox Adaptive Controller (**P3**$_P$, **P7**$_{PC}$, **P9**$_{PC}$, **P10**$_{PC}$), custom buttons and joysticks (**P7**$_{PC}$, **P9**$_{PC}$, **P10**$_{PC}$), and software for remapping controller and keyboard keys (**P1**$_P$, **P8**$_{PC}$). No participants had previous experience with partial automation.

3.2 Procedure

The interviews were conducted online, using video-calling platforms, and were audio-recorded to facilitate the following analysis. In addition to the participant, at least two members of the research group were always present during each interview: one with the role of main interviewer and the other with a support function, tasked with intervening with additional questions not initially planned. Each interview lasted between 60 and 90 minutes. The initial interview outline[4] drew inspiration from prior works on shared control [20–22, 37, 80]. The first part of each interview was dedicated to human cooperation technologies. With pilots and copilots, we explored how they used these technologies, which ones they preferred, and what advantages and limitations they had. With accessibility experts, we investigated how they supported users who come seeking a new game setup tailored to their needs. We then proceeded to understand how they integrate human cooperation technologies into these setups.

In the second part, a concept of a partial automation system for video game accessibility was presented, and interviewees were asked for their opinion, how they would use the system, and what requirements it should have. The system was described as a shared control solution, similar to *Xbox Controller Assist* [60], but designed to allow playing without the need for a second person. The users would be able to select which actions they wanted to control, while the system would autonomously control the remaining ones. The description of the system was purposefully kept vague to encourage the elicitation of its potential functionalities. The initial questions were expanded based on the outcome of the first two interviews. In particular, two additional topics on human cooperation were introduced: communication between the pilot and copilot during gameplay and reasons for not using this technology.

3.3 Data Analysis

The audio recordings of the interviews were transcribed and analyzed using the reflexive thematic analysis methodology [89]. The first interview was jointly analyzed by four researchers, who identified initial codes. Subsequent interviews were randomly divided among the researchers. Each interview was coded by one researcher and reviewed by a second one, who integrated additional observations and annotated any ambiguities or alternative interpretations. In a subsequent meeting, all codes were reviewed jointly by the entire team. The reported disagreements and ambiguities were systematically discussed, and the relevant parts of the transcriptions were re-examined until the plurality of views was adequately represented and documented. Finally, the extracted codes were consolidated and organized into sub-themes and main themes through an iterative process of reflection and discussion among researchers.

[4]The initial interview outline is available as supplemental material.

4 Results

Six main themes were identified (Table 2): *Shared Control Benefits and Limitations* (Section 4.1), *Human Cooperation Limitations Addressed in Partial Automation* (Section 4.2), *Copilot's Interventions* (Section 4.3), *Negotiating Collaboration* (Section 4.4), *Interaction* (Section 4.5), and *Factors Affecting the Collaboration* (Section 4.6). Note that, although the interviews focused on two distinct areas — human cooperation technologies and the proposal of a partial automation system — we unify their analysis, treating the two areas orthogonally with respect to the identified themes when possible. This choice is motivated by the similarity of the collected codes, which are largely transversal to both areas.

4.1 Shared Control Benefits and Limitations

This section explores the benefits of shared control as an accessibility tool (Section 4.1.1), its impact on sociality and inclusion (Section 4.1.2), and ethical concerns related to its misuse (Section 4.1.3).

4.1.1 Accessibility.

Human cooperation. Human cooperation enables people with disabilities to potentially play any game that is not accessible using other assistive technologies (**P2**$_P$, **P3**$_P$, **P5**$_E$, **P7**$_{PC}$, **P8**$_{PC}$, **P9**$_{PC}$, **P10**$_{PC}$). It is also beneficial for those games that can be played autonomously, but at a cost of a high physical or cognitive load (**P7**$_{PC}$). Indeed, accessibility experts (**P5**$_E$, **P6**$_E$, **P13**$_E$, and **P14**$_E$) confirm that human cooperation is indeed useful as an assistive solution:

> **P5**$_E$: *I use [human cooperation] all the time. I know that, if somebody won't be able to use two [gamepad sticks], I almost always will have [human cooperation] as an option.*

Partial automation. The idea of partial automation as an alternative to human cooperation was acclaimed by most participants, who recognize its potential as an assistive technology. For example, **P1**$_P$ considers that, similarly to human cooperation, partial automation would be useful to access games too complex to be played autonomously. **P2**$_P$, **P7**$_{PC}$, **P8**$_{PC}$, **P9**$_{PC}$, and **P10**$_{PC}$ furthermore highlight that partial automation could be particularly useful in multiplayer games, which commonly have fewer accessibility options. **P7**$_{PC}$ and **P10**$_{PC}$ would be eager to test this solution. For **P5**$_E$, partial automation could also be effective for users with very low mobility: in such cases, users could delegate most interactions to the system, while still having a decisional role on the game progress. That would make the gaming experience more similar to an interactive movie, preserving the narrative content and the active role of the player.

Table 2: Themes and sub-themes identified through the reflexive thematic analysis

Theme	Sub-themes
1. *Shared Control Benefits and Limitations*	1. *Accessibility* 2. *Sociality and Feeling of Inclusion* 3. *Ethical Concerns*
2. *Human Cooperation Limitations Addressed in Partial Automation*	1. *Loss of Autonomy* 2. *Copilot Availability* 3. *Copilot's Engagement During Play*
3. *Copilot's Interventions*	1. *Assistance During Game Setup* 2. *Assistance with Menu Access* 3. *Assistance by Playing* 4. *Assistance by Signaling* 5. *Assistance by Suggesting*
4. *Negotiating Collaboration*	1. *Actions Separation* 2. *Policies Guiding Action Assignment* 3. *Leadership Management* 4. *Copilot's Operational Autonomy*
5. *Interaction*	1. *Verbal and Non-Verbal Communication* 2. *Intent Understanding*
6. *Factors Affecting the Collaboration*	1. *Knowledge of the Game* 2. *Relationship Between Pilot and Copilot*

4.1.2 Sociality and Feeling of Inclusion.

Human cooperation. Nine participants (**P3**$_P$, **P5**$_E$, **P6**$_E$, **P7**$_{PC}$, **P8**$_{PC}$, **P9**$_{PC}$, **P10**$_{PC}$, **P12**$_{CE}$, **P14**$_E$) pointed out that human cooperation is not simply about being able to play, but also about socializing through playing together. Thus, **P5**$_E$ strongly advocates for human cooperation, even when other accessibility options are available:

> **P5**$_E$: *I think that there is a lot of good psychosocial component to playing together. And when you play alone, it's ok... It's just I think it's better with others.*

P14$_E$ noted that playing through human cooperation can also strengthen the bond with the copilot, for instance, when the latter is a family member. That's why, when working with children, **P14**$_E$ usually recommends human cooperation as a solution. Furthermore, **P7**$_{PC}$, **P8**$_{PC}$, **P9**$_{PC}$, and **P10**$_{PC}$, who participate together in weekly gaming workshops (Section 3.1), reported that these meetings are also an opportunity to socialize. **P3**$_P$ and **P14**$_E$ further explained that human cooperation enables them to discuss with friends about games they would otherwise be unable to access. Without human cooperation, **P3**$_P$ would be limited to watching others play, which could lead to feelings of exclusion. According to **P4**$_C$, human cooperation can also be used to introduce beginners to gaming. For example, allowing a parent to learn to play alongside their children, thus fostering family bonding.

Partial automation. Social interaction with the human copilot is lost when partial automation is used. For this reason, **P3**$_P$ would still prefer to play with **P4**$_C$ via human cooperation, even if a partial automation system were available, as those moments provide an opportunity to spend time together. **P7**$_{PC}$ and **P12**$_{CE}$ agree on this aspect, expressing that they would not want to abandon human cooperation entirely to avoid losing the social benefits it entails. In contrast, **P1**$_P$, **P11**$_{CE}$, and **P14**$_E$ observed that partial automation does not necessarily remove the social dimension; rather, it may create new opportunities. Indeed, partial automation could enable players to independently access multiplayer games, which by their nature foster social interaction and group play. For example, **P11**$_{CE}$ and **P12**$_{CE}$ imagined forming a team of players using partial automation to play together in multiplayer games. Similar to human cooperation, partial automation could also facilitate the inclusion of new players in gaming (**P1**$_P$).

4.1.3 Ethical Concerns.

Some accessibility solutions can be misused or exploited for cheating, undermining the intended gameplay and negatively affecting other players' experiences. For this reason, multiplayer games may restrict their use.

Human cooperation. **P4**$_C$, **P5**$_E$, and **P14**$_E$ note that human cooperation systems are subject to this issue. According to **P2**$_P$, the problem is exacerbated by the fact that multiplayer games are generally less accessible than single-player games and would therefore need better accessibility support.

Partial automation. A partial automation system would face similar limitations. For this reason, **P6**$_E$ is skeptical about its applicability in multiplayer contexts. **P2**$_P$ suggests distinguishing between competitive and non-competitive multiplayer games: while in the former the use of partial automation should be regulated, in the latter it should remain unrestricted, as in single-player games. To address this concern, **P12**$_{CE}$ proposes the development of partial automation systems capable of learning and adapting to the player's abilities, thereby moderating the level of assistance provided. This aspect is discussed in more detail in Section 4.3.3. Overall, **P5**$_E$ expects mixed reactions:

> **P5**$_E$: *The people who need [partial automation] would celebrate, the people who don't need it would think it's cheating. I mean, some of them, not all of them. I am able to play, but I would celebrate because I know that so many of the people that I've worked with will now be able to play more.*

4.2 Human Cooperation Limitations Addressed in Partial Automation

Human cooperation also has limitations that affect its widespread adoption: reduced pilot's autonomy (Section 4.2.1), need for a copilot (Section 4.2.2), and concerns about the copilot's engagement during play (Section 4.2.3).

4.2.1 Loss of Autonomy.

Human cooperation. Human cooperation implies dependence on another person, which limits the player's autonomy. Participants react to this limitation in different ways. For **P2**$_P$, who is blind, no other accessibility solution is viable. While not uncomfortable requesting help, **P2**$_P$ acknowledges that dependence can limit play options. In contrast, **P1**$_P$ avoids human cooperation despite other tools being less effective. This is in part due to logistical barriers, and in part because having to ask for help every time would feel burdensome. **P13**$_E$ confirms that this reluctance is particularly common among people with acquired disabilities, possibly due to the inability to regain pre-disability skill levels:

> **P13**$_E$: *There will always be people who, if they can't have full access to the game they want to be able to play, would choose not to play. We have definitely met people that would say: "If I can't do this myself completely independently, I'd rather not do it.". [...] I think most people would want to be able to play the way that they played before, and so it might not be as enjoyable if they don't have full access.*

P2$_P$ reports being criticized by other players with visual impairments for relying on the assistance of another person, believing that this misrepresents the gaming experience of visually impaired players. In general, **P5**$_E$, **P6**$_E$, **P13**$_E$, and **P14**$_E$ prioritize finding game configurations preserving independence, without having to rely on human cooperation.

Partial automation. Partial automation eliminates the dependency on another person, offering greater autonomy to the player. For example, if such a solution were available, **P2**$_P$ would attempt higher difficulty levels, replay games multiple times, and commit to longer sessions without the feeling of weighing on someone else. For this reason, **P6**$_E$ explained that, if partial automation were available, it would be a *second-line option*, adopted when no complete configuration based on traditional accessibility solutions can be identified. Human cooperation would therefore be considered only a *third-line option*, used solely in the absence of partial automation. Similarly, **P8**$_{PC}$ and **P9**$_{PC}$ highlighted that the independence afforded by partial automation constitutes a clear advantage over human cooperation approaches.

4.2.2 Copilot Availability.

Human cooperation. Human cooperation is also limited by the copilot's availability. One side of the problem is logistical, since the pilot and the copilot need to be physically co-located to play. All participants reported that this requirement is a significant constraint in the use of human cooperation, as it drastically reduces the pool of potential copilots. A common choice is to have a family member as the copilot, which is what **P3**$_P$ and **P4**$_C$ do. The relationship between the pilot and the copilot is discussed in Section 4.6.2. Additionally, the copilot may not always have time to play (**P3**$_P$). To overcome the challenges of copilot's availability, some participants arrange gaming sessions in shared physical spaces. For example, for **P7**$_{PC}$, **P8**$_{PC}$, **P9**$_{PC}$, and **P10**$_{PC}$, who meet at weekly gaming workshops, finding a copilot is straightforward, but only during scheduled sessions. An alternative, known only to **P2**$_P$, is remote human cooperation (Section 2.3.1). However, this approach requires a complex configuration of third-party hardware and software, and it introduces interaction latency that complicates the coordination. **P2**$_P$ notes that such a solution, if natively integrated in gaming platforms, would make it easy to meet copilots, thus improving game accessibility for many people with disabilities. When no copilot is available, fallback strategies vary. **P2**$_P$, **P8**$_{PC}$, and **P10**$_{PC}$ stop playing altogether; **P3**$_P$, **P7**$_{PC}$, and **P9**$_{PC}$ switch to more accessible alternatives, such as mobile games. Finally, **P2**$_P$, **P9**$_{PC}$, and **P10**$_{PC}$ sometimes play with different copilots, which introduces the challenge of learning to play together (Section 4.6.2).

Partial automation. Partial automation doesn't have copilot availability issues. That's why **P3**$_P$ would find a system based on it "fantastic" as it would allow playing even when no one is available as a copilot. For **P2**$_P$, this approach would benefit the copilots as well, by freeing them from the time commitment of learning and playing the entire game:

> **P2**$_P$: *You don't have to wait for a person to be available, you can play any game at any time on your own terms without having to rope a second person in and make them take hundreds of hours of their life.*

4.2.3 Copilot's Engagement During Play.

Human cooperation. **P1**$_P$ notes that the gaming experience in human cooperation is not fully comparable to that offered by collaborative multiplayer games. In the latter, the roles of the two players are generally equivalent or equally significant, whereas in human cooperation, the division of tasks is often unbalanced. According to **P1**$_P$, this imbalance can affect the copilot's enjoyment, as they are frequently assigned only a supporting role. Indeed, **P1**$_P$ considers playing as a copilot to be a predominantly negative experience, describing it as an activity performed mainly to help someone else rather than for personal enjoyment. It should be noted, however, that **P1**$_P$ has always acted as a pilot and has never directly experienced the role of copilot. In contrast, **P4**$_C$ reported finding their experience as a copilot rewarding, viewing it also as an opportunity to discover new video games.

Partial automation. With partial automation, this issue does not arise, as the copilot's role is replaced by software.

4.3 Copilot's Interventions

The copilot can support the pilot during gameplay in various ways: assistance with the setup before starting to play (Section 4.3.1), support with menu access (Section 4.3.2), direct control of commands during gameplay (Section 4.3.3), signaling elements of interest in the game (Section 4.3.4), and suggestions on how to proceed in the game (Section 4.3.5).

4.3.1 Assistance During Game Setup. Setting up a game configuration is a long and complex process, which in some cases can lead to frustration (**P13**$_E$) or even to giving up playing (**P11**$_{CE}$), as also noted in prior literature [58]. The presence of a copilot can facilitate these operations.

Human cooperation. Some participants are only able to play if another person helps them to prepare the gaming hardware and software setup (**P3**$_P$, **P8**$_{PC}$, **P9**$_{PC}$, **P10**$_{PC}$, **P13**$_E$, **P14**$_E$). Support can sometimes be needed even for those who can afterward play independently (**P5**$_E$, **P6**$_E$, **P11**$_{CE}$). For example, the player may be able to play independently using an accessible controller, but may need support to connect, position, and configure the controller itself. **P5**$_E$ and **P6**$_E$ always try to maximize the user's autonomy, but some users still request their help in these situations.

Partial automation. While none of the participants commented about the assistance during setup in partial automation, we note that a software copilot cannot physically assist the pilot in preparing the gaming peripherals' configuration before starting to play. Thus, in this case, human support may still be required to start the game.

4.3.2 Assistance with Menu Access. Enabling in-game accessibility options through the game menu is sometimes needed to make a game accessible (**P2**$_P$, **P5**$_E$, **P6**$_E$, **P11**$_{CE}$, **P12**$_{CE}$, **P13**$_E$). However, the menus themselves may lack accessibility (**P2**$_P$, **P11**$_{CE}$, **P12**$_{CE}$), and solutions used to make the active gameplay accessible often do not work for accessing menus (**P12**$_{CE}$). To access text from game menus and navigate them **P2**$_P$ uses Optical Character Recognition tools. This introduces additional workload and, as **P2**$_P$ points out, these technologies often make mistakes.

Human cooperation. Due to the above problem, **P2**$_P$ highlights that having a copilot ready to help with menu access is much preferable. **P11**$_{CE}$ and **P12**$_{CE}$ also point out the issue of menu accessibility, explaining that people participating in their workshop very rarely manage to start a gaming session without help from a copilot:

> **P12**$_{CE}$: *If games were designed with accessibility in mind from the start, with continuous scrolling menus where you only need to click on the options, that would be one thing. But menu accessibility is rarely implemented, even in accessible games. It may seem trivial, but it's extremely limiting for our user base.*

For example, **P11**$_{CE}$ described the difficulty of navigating menus in *Forza Horizon* [76] due to the fact that horizontal scrolling was not configured on the eye tracker employed by the user, as it was not needed during the game itself.

Partial automation. **P11**$_{CE}$ and **P12**$_{CE}$ highlight that partial automation should also support menu navigation.

4.3.3 Assistance by Playing.

Human cooperation. Assistance by controlling some of the game actions is usually the main form of support by the copilot. **P1**$_P$, **P3**$_P$, **P7**$_{PC}$, and **P8**$_{PC}$ all play with a copilot who primarily assists them by controlling actions in the game. For example, when playing *Call of Duty* [4], **P1**$_P$'s copilot controls the camera direction, reloads the weapon, and launches grenades. **P1**$_P$ controls all other actions.

Partial automation. All participants envision partial automation assisting them by directly controlling some of the game actions, similar to a human copilot. However, some expressed doubt about the copilot's ability to balance the level of support provided, avoiding being too skilled compared to the pilot and thus undermining the user experience (**P1**$_P$, **P5**$_E$, **P11**$_{CE}$, **P12**$_{CE}$). In this sense, **P11**$_{CE}$ and **P12**$_{CE}$ expect partial automation to bring the player to "the same level as others", prioritizing enjoyment over in-game success. According to **P5**$_E$, the main goal should be to enable the pilot to reach a minimum standard that ensures enjoyment, without completely replacing their skill. **P11**$_{CE}$ and **P12**$_{CE}$ also suggested that the level of support could be dynamically adjusted, progressively decreasing as the pilot improves their performance. If this is not possible, **P4**$_C$ and **P11**$_{CE}$ would at least like to manually select the desired level of support.

4.3.4 Assistance by Signaling.

Human cooperation. Another way of assisting the pilot is by signaling what's on the screen and what is happening in the game. This support is particularly valuable to **P2**$_P$, who is blind. Indeed, **P2**$_P$ reports that their copilot must clearly indicate what is happening and which actions they are performing, so that they can coordinate effectively. For instance, in *God of War* [81], **P2**$_P$ asks the copilots to frequently report their own and their opponent's health status.

Partial automation. **P2**$_P$ states that partial automation should also provide this type of support, describing what is happening in the game rather than giving instructions on what to do. This way, the player can make their own decisions on how to act.

> **P2**$_P$: *If you give me that information, I can do what most players would be able to do. I'm going to decide who I target first. I'm going to decide how I play. Whereas if you're just controlling me, I'm like "Yeah, I can shoot when you tell me to shoot, but where's the fun in that?".*

4.3.5 Assistance by Suggesting. Finally, the copilot can support the pilot by providing suggestions. This is particularly useful when the pilot is struggling to make progress in the game.

Human cooperation. **P11**$_{CE}$ and **P12**$_{CE}$ provide this type of support in their weekly gaming group. However, both note that, when giving suggestions, they always try to guide the pilot towards reasoning, without fully solving their problem. This happens, for example, when assisting their pilot while playing *Inside* [75], a puzzle-platform game:

> **P11**$_{CE}$: *I tell them "Let's go back, maybe we missed something" and when we go back, they notice on their own something they didn't see earlier, like a handle or something to pull. Then they can go on by themselves.*

Partial automation. In partial automation **P2**$_P$ would like a similar support, where the copilot does not intervene in the game but provides suggestions. Indeed, **P2**$_P$ notes that it is preferable for the copilot not to intervene directly to preserve the pilot's sense of agency. **P11**$_{CE}$ also imagines that the system could detect when the player is stuck in the game and give them suggestions to help them understand how to proceed. For **P9**$_{PC}$ and **P11**$_{CE}$, it is important that suggestions do not directly reveal the solution, to leave the player with the satisfaction of solving the puzzle themselves. Finally, some participants suggest that the copilot could initiate communication proactively, for example, by recognizing moments of difficulty and spontaneously offering their help (**P1**$_P$, **P2**$_P$, **P12**$_{CE}$).

4.4 Negotiating Collaboration

Participants discussed the distribution of the game actions between the pilot and the copilot (Section 4.4.1), the criteria for the assignment of actions (Section 4.4.2), how they manage leadership in decision-making (Section 4.4.3), and the level of the copilot's autonomy in intervening during the game.

4.4.1 Actions Separation. The division of actions between the pilot and the copilot can be static, without changes during a game, or dynamic. It can be decided before starting, or while the game is in progress. Finally, actions can be separated strictly, with each player controlling a subset of actions, or overlapping, with some actions controlled by both.

Human cooperation. In general, **P1**$_P$ notes that the division varies depending on the game. However, **P3**$_P$ specifies that, when playing games belonging to the same genre, they often use similar configurations. The division of actions is often defined before starting to play. For example, **P1**$_P$ discusses the division with their copilot, since, having prior shared gaming experience, the copilot understands well what **P1**$_P$ is better at. Only **P2**$_P$ starts playing without a clear division of actions, preferring to work out together with the copilot what each of them does best as they play. Usually, the action separation is static throughout the game. **P2**$_P$, instead, adopts a dynamic separation during the game: in specific sections that they would not be able to manage alone, the pilot hands over full control of the game to the copilot. Most participants strictly separate the controls managed by the pilot and those managed by the copilot (**P1**$_P$, **P2**$_P$, **P3**$_P$, **P4**$_C$, **P7**$_{PC}$, **P8**$_{PC}$, **P9**$_{PC}$, **P10**$_{PC}$). For example, **P1**$_P$ and **P3**$_P$ generally delegate full camera control to the copilot:

> **P4**$_C$: *One of* ***P3***$_P$*'s biggest weaknesses actually is camera control on the right stick: using the camera while doing everything else is very, very hard. So, I'd say the thing I do most is camera control. [...] I'll point at an enemy and* ***P3***$_P$ *will go ahead and handle the job.*

While strict control separation is more common, overlap between the pilot's and the copilot's controls is also possible. This occurs mostly in family contexts or among children, where it becomes unclear who does what (**P6**$_E$). However, as **P13**$_E$ points out, the lack of clarity on who is responsible for which controls can lead to confusion and frustration.

Partial automation. In partial automation, participants also expect the ability to customize which actions to automate and how they want to be supported. For example, **P3**$_P$, **P4**$_C$, **P5**$_E$, and **P6**$_E$ would prefer to divide the actions between pilot and copilot before starting to play, just as they do in human cooperation. **P3**$_P$ assumes that the configuration would be similar to the one they already use when playing with a human copilot, delegating camera control to the software. Instead, **P1**$_P$ would experiment with different configurations to find the one that best suits their needs. **P3**$_P$ also hypothesizes the possibility of a dynamic division, as they would need the copilot's assistance only in specific sections of the game. Finally, **P5**$_E$ suggests asking the software copilot to propose how to divide the actions. **P1**$_P$ expresses a similar view but emphasizes that the copilot should not be overly intrusive in suggesting configuration changes or disrupting the flow of the game.

> **P5**$_E$: *One of the things that is gonna be very important is that you understand the player, the person that you're building it for. [...] You might try to find some way to do an evaluation on the person [...], and then the AI could read what the evaluation says and say "I think I already know what you need. You need the right trigger, the right bumper acceleration, and a boost".*

4.4.2 Policies Guiding Action Assignment.

Human cooperation. We identified three main policies guiding the allocation of controls between pilot and copilot. First, the pilot's abilities are taken into account (**P1**$_P$, **P7**$_{PC}$, **P8**$_{PC}$, **P9**$_{PC}$, **P10**$_{PC}$), assigning to the copilot those controls that are less accessible to the pilot. For example, **P1**$_P$ has difficulties in controlling the left hand, and therefore remaps most frequently used game actions to buttons on the right side of the controller. The remaining actions are then assigned to the copilot. When both the pilot and the copilot have disabilities, the allocation is based on the abilities of both. For instance, **P7**$_{PC}$, **P8**$_{PC}$, **P9**$_{PC}$, and **P10**$_{PC}$ all have a disability and regularly play in pairs using human cooperation, supporting each other during the game. Similarly, **P5**$_E$ reports having applied human cooperation multiple times with two people who both had disabilities. Second, as hinted by **P1**$_P$, the most important game actions are assigned to the pilot to ensure that they maintain a sense of agency over the game. For example, **P3**$_P$, as a pilot, controls most commands, while **P4**$_C$ assists with camera control, which **P3**$_P$ cannot manage simultaneously with other actions. Third, participants divide actions based on the macro-functionality they control. For example, **P7**$_{PC}$ and **P8**$_{PC}$ separate movement from other actions, assigning all movement-related actions to one player and the remaining ones to the other.

Partial automation. The first two policies were also mentioned in relation to partial automation. First, participants suggested delegating to the software copilot those actions that the pilot cannot control or would struggle to control. For example, **P3**$_P$ noted that partial automation could handle complex input sequences or those requiring quick execution. Similarly, actions requiring high precision, such as aiming (**P3**$_P$, **P9**$_{PC}$), could also be delegated to the copilot. In this context, camera orientation was mentioned by participants as a challenging action to manage and one that could be delegated to the copilot (**P1**$_P$, **P3**$_P$, **P7**$_{PC}$). Second, participants suggested delegating secondary actions to the copilot, in particular those used only occasionally (**P1**$_P$, **P3**$_P$), while leaving the main controls to the pilot.

4.4.3 Leadership Management.

Human cooperation. In general, the person with disabilities acts as the pilot and makes game-wide strategic choices (**P2**$_P$). However, when both players have a disability, the definition of who is the pilot and who is the copilot becomes blurred. For example, **P7**$_{PC}$, **P8**$_{PC}$, **P9**$_{PC}$, and **P10**$_{PC}$ all have disabilities and play in pairs through human cooperation (Section 4.4.2). Every decision is therefore made collaboratively, and both players contribute equally to the game. Finally, **P5**$_E$ notes that sometimes a person without disabilities can be the pilot, with a person with disabilities as the copilot:

> **P5**$_E$: *Another patient that I had who also had hemiplegia [...] wanted to play Call of Duty [4] with his son. And so [...] his son would look around and aim, and the father would just shoot.*

Leadership can also be temporarily assigned to the copilot. **P2**$_P$ does this in game sections that require complex or rapid input sequences. **P5**$_E$ notes that when the copilot takes the lead, they must have experience in the specific game (Section 4.6.1), otherwise the experience could be frustrating for the pilot.

Partial automation. None of the participants mentioned the possibility of sharing or leaving game-wide strategic leadership to the software copilot. Indeed, they believe that the pilot should maintain the leadership and partial automation should intervene only when the pilot is in difficulty, and generally avoid compromising their sense of agency (**P1**$_P$, **P3**$_P$, **P6**$_E$). **P3**$_P$ envisions leaving temporary control to the software copilot in specific contexts, as **P2**$_P$ does in human cooperation, for example, when the game requires action execution speed that is too high for them:

> **P3**$_P$: *If the AI could change depending on the situation. I think that would be nice 'cause I hope I don't need the same level of help for an entire game. There are certain scenes that I need more help with than others.*

4.4.4 Copilot's Operational Autonomy. The copilot's level of operational autonomy exists on a spectrum. At the one extreme, the copilot can act as an actuator, following step-by-step instructions from the pilot, who indicates when to act and which buttons to press. At the other extreme, the copilot can be completely autonomous, executing actions based on a set of rules agreed upon with the pilot, without requiring specific instructions.

Human cooperation. In human cooperation, **P1**$_P$ is the only one who sometimes plays with a copilot acting solely as an actuator. This is necessary when the copilot is not experienced with the game (Section 4.6.1), as reported by **P13**$_E$. Instead, when the copilot is familiar with the video game, they are able to act autonomously. This results in the copilot providing more effective help, which causes the pilot to feel more immersed in the game (**P2**$_P$). As a result, most participants play with a copilot that acts autonomously (**P1**$_P$, **P3**$_P$, **P4**$_C$, **P5**$_E$, **P6**$_E$, **P7**$_{PC}$, **P8**$_{PC}$, **P9**$_{PC}$, **P10**$_{PC}$).

Partial automation. In partial automation, **P2**$_P$, **P3**$_P$, **P4**$_C$, **P9**$_{PC}$, **P11**$_{CE}$, and **P12**$_{CE}$ envision playing with a copilot that only follows their instructions. **P9**$_{PC}$ would use this support modality especially in logic games, where a more autonomous copilot might reveal the solution to proposed puzzles. **P7**$_{PC}$ and **P8**$_{PC}$, instead, would play with a copilot that autonomously controls a predefined set of actions. **P1**$_P$ and **P5**$_E$ also expect that the copilot can act autonomously, but emphasize the importance of defining rules that precisely establish what it should and can do, for example, preventing the copilot from wasting ammunition in a first-person shooter game.

4.5 Interaction

The interaction between pilot and copilot in shared control is fundamental for exchanging information and coordinating during gameplay. This interaction is based both on communication, which can be verbal or non-verbal (Section 4.5.1), and on the ability to understand each other's intentions (Section 4.5.2).

4.5.1 Verbal and Non-Verbal Communication.

Human cooperation. All participants primarily interact through verbal communication, engaging in dialogue with the other player to provide instructions and decide the strategy to adopt. However, the way pilots and copilots communicate often evolves over time. For example, by always collaborating with the same copilot, **P2**$_P$, **P7**$_{PC}$, and **P8**$_{PC}$ have developed short and quick commands over time to easily communicate with their copilot. They all believe that this way of communicating is more efficient and less tiring than using full sentences. Indeed, **P2**$_P$ recounts how their copilot acted as a guide in *God of War* [81], using synthetic commands like "clear left" or "swoop" to indicate actions to perform in combat. This way, **P2**$_P$ managed to complete some particularly complex game sequences requiring direct copilot intervention only in rare cases. For **P9**$_{PC}$ and **P10**$_{PC}$, this type of communication is particularly important in faster-paced games for real-time coordination, while in slower games the communication is more articulate and focused on making decisions jointly. Only **P7**$_{PC}$ also relies on non-verbal communication through eye contact, which they find more accessible due to speech difficulties.

Partial automation. Similarly, in partial automation, some participants propose using short and quick commands to give instructions to the copilot (**P7**$_{PC}$, **P8**$_{PC}$). However, **P7**$_{PC}$ also highlights possible difficulties related to voice use, which could slow down interaction and, for those with speech problems, be less accessible. For this reason, **P7**$_{PC}$ suggests teaching personalized commands to the software copilot. Alternatively, **P1**$_P$ suggests the use of brain-computer interfaces, which would enable immediate communication with the pilot, particularly suitable for fast-paced games.

4.5.2 Intent Understanding.

Human cooperation. In human cooperation, the copilot's understanding of the pilot's intentions occurs primarily through explicit requests (**P6**$_E$). With experience, however, the copilot learns to anticipate the pilot's needs, becoming progressively more effective in providing support by observing how the pilot plays (**P2**$_P$). Similarly, the pilot's understanding of the copilot's intentions is primarily based on verbal communication (**P2**$_P$, **P3**$_P$, **P4**$_C$, **P5**$_E$).

Partial automation. In partial automation, mutual understanding of intentions remains equally important (**P5**$_E$). If the copilot fails to understand the pilot's objective, automatic intervention might be perceived as arbitrary. **P12**$_{CE}$ and **P13**$_E$ agree on this issue, but also express doubts about the reliability of an automatic system in non-linear games, where possible actions are multiple and less predictable. If, instead, the pilot fails to coordinate with the copilot, they might not be able to anticipate the copilot's actions and perceive a reduced sense of agency over the game (**P5**$_E$):

> **P5**$_E$: *I think it would be tricky. If I had a shooting game and I wanted [the AI] to shoot for me, because I can't physically shoot the buttons. How do you control it from not using all of your ammunition? Will it just shoot all the time? Is there something that says "Ok, now I have targeted my person, and now you shoot"?*

In this context, **P12**$_{CE}$ proposes a dynamic interaction model, in which the copilot analyzes the pilot's interactions and commands, proposing interventions consistent with the player's intentions, and improving the ability to anticipate their needs over time.

4.6 Factors Affecting the Collaboration

The collaboration between the pilot and the copilot is influenced by several factors that determine its effectiveness. We first analyze how the copilot's knowledge of the game impacts the collaboration (Section 4.6.1). Then we examine the impact of the relationship between the pilot and the copilot on in-game collaboration (Section 4.6.2).

4.6.1 Knowledge of the Game.

Human cooperation. Participants emphasized that the copilot should have at least a general understanding of video games and their common mechanics. **P5**$_E$, **P6**$_E$, and **P11**$_{CE}$ consider this level of familiarity sufficient to provide effective support. **P4**$_C$ and **P6**$_E$ also mentioned specific mechanics that the copilot should be able to handle, such as camera control, character movement, and menu navigation. For this reason, **P5**$_E$ includes a training phase for the copilot when proposing human cooperation to players with disabilities. In general, knowledge of the specific game for which support is provided is not deemed essential, but it can improve the gaming experience, particularly in complex video games (**P5**$_E$, **P13**$_E$, **P14**$_E$). For this reason, **P4**$_C$ independently learns the game mechanics before assisting **P3**$_P$ as copilot. **P5**$_E$ also considers experience important to facilitate communication during gameplay. For **P1**$_P$, however, thorough knowledge of the game is essential when the copilot assumes a guiding role and needs to provide suggestions on how to proceed (Section 4.3.5). In such cases, a lack of experience with the specific game may severely undermine the effectiveness of the assistance (**P5**$_E$), generating frustration in the pilot, who is unable to progress further. **P2**$_P$ needs the copilot to describe what is happening on screen and therefore highlights that an experienced copilot better understands the state of the game and consequently communicates it more effectively:

> **P2**$_P$: Let's say you get a person who knows a game well and one who doesn't know it as well. Like, one knows the enemy types and the other doesn't. [...] If my copilot's saying "Okay, there's a terminator[5] in the distance", I'm like "Okay, I know what to do" [...] Whereas if the person doesn't know the enemies as well you then have a sense of "Oh God, this is terrifying. I don't know what's happening".

[5] *Terminator* is an enemy in the video game *Warhammer 40,000: Space Marine 2*

Partial automation. **P1**$_P$ argues that a software copilot should be trained for each video game specifically to be able to provide effective support. However, **P1**$_P$ also questions if developing a copilot capable of adapting to different games is even feasible.

4.6.2 Relationship Between Pilot and Copilot.

Human cooperation. Participants reported that the copilot is often a family member or a friend. For example, **P3**$_P$ is a family member and caregiver of **P4**$_C$. Due to this, **P3**$_P$ is available to play together most of the time, and dedicates significant effort to better support **P4**$_C$ (*e.g.*, by trying a game in advance before playing together). Furthermore, this type of relationship can facilitate communication, reduce logistical difficulties (Section 4.2.2), and improve coordination during play. For **P1**$_P$, **P7**$_{PC}$, **P8**$_{PC}$, **P9**$_{PC}$, and **P10**$_{PC}$ a friend acts as copilot. **P1**$_P$ highlights that prior gaming experience shared together facilitates collaboration. For these reasons, experts (**P5**$_E$, **P6**$_E$, **P13**$_E$, **P14**$_E$), when recommending the use of human cooperation, suggest involving someone familiar to the pilot. Familiarity is particularly important in fast-paced and complex video games (**P3**$_P$, **P7**$_{PC}$, **P8**$_{PC}$), whereas it is less critical in simpler games (**P9**$_{PC}$, **P10**$_{PC}$). Furthermore, **P6**$_E$ notes that some of the people they assist are reluctant to accept help from someone who doesn't already support them in daily life. In other cases, a copilot can also be a stranger met online. This option was mentioned by **P2**$_P$, who also observed that this type of copilot selection can lead to variable outcomes, depending on the compatibility with the copilot:

> **P2**$_P$: *There's this idea of being drift compatible[6] where you are in this flow state, you are both sort of synced up with each other, and it's all about shared communication.*

P2$_P$ further notes that each new collaboration requires time to build the coordination needed for a smooth and satisfying gameplay experience. Indeed, the first hours of play with a new copilot are dedicated to deciding how to play together (Section 4.4), communicating (Section 4.5), and building a relationship. During this process, errors and incomprehension are possible. Thus, **P2**$_P$ and **P5**$_E$ highlight that being positive and tolerant is important to avoid frustration.

Partial automation. As **P1**$_P$ pointed out, the software copilot should adapt its assistance to the player's skill level. Otherwise, it can make the player dependent on the provided support, hindering autonomous growth and limiting their ability to experiment and find solutions independently. Similarly, **P11**$_{CE}$, and **P12**$_{CE}$ believe that the copilot should learn the pilot's play style over time and adapt to it, as generic support may not be adequate or effective.

5 Discussion

Prior works highlight that shared control can be used to access games that cannot be made accessible through existing accessibility solutions [19, 36]. Our research confirms this (Section 4.1) and also uncovers which types of support should be provided and how (Section 5.1). Additionally, we discuss how technologies supporting

[6] *Drift compatible*: Term from the film Pacific Rim [26] in which two pilots jointly control a robot and need to be compatible to perform effectively

shared control should promote autonomy and sociality (Section 5.2). Then, we discuss how the collaboration between the pilot and the copilot could translate into partial automation (Section 5.3), and how to facilitate communication and mutual intent understanding during gameplay (Section 5.4). The resulting design guidelines and recommendations are summarized in Table 3. Finally, we discuss the generalizability of these guidelines (Section 5.5) and report limitations of our study (Section 5.6).

5.1 Supporting Different Forms of Assistance

The participants described three main ways in which the software copilot could assist them: by controlling the game actions they were unable to perform themselves (Section 5.1.1); by providing additional information about the game, what happened in it, and how to play it (Section 5.1.2); and by assisting with the game setup before playing (Section 5.1.3).

5.1.1 Delegate control of some game actions. Prior works motivate the use of partial automation for game accessibility, arguing that: "Games may require inputs that some players cannot provide with any device." [19]. Our analysis confirms this, since participants reported difficulties with **fine control of input intensity [G1.1]**, for example, when aiming in first-person shooters or steering in driving games (Section 4.4.2). However, using existing accessibility tools and control remapping, most participants are able to mitigate this problem, and most of them can actually activate any single control individually. Instead, our participants found it difficult to control multiple actions concurrently or in rapid succession. Borrowing a term from the machine learning domain [48], we call this issue the *Curse of Dimensionality*. Indeed, multiple participants reported difficulties when **simultaneously maneuvering multiple directional controls [G1.2]**, for example, the movement and the camera (Section 4.4.1). The curse of dimensionality problem is further exacerbated when a large number of controls need to be managed during **fast-paced interaction sequences [G1.3]** (Section 4.4.3). Human cooperation is an effective solution to address this problem since it reduces the number of controls assigned to the pilot by delegating some to the copilot. For the same reason, participants noted that partial automation could be similarly effective, highlighting its potential to replace human cooperation. For example, in a first-person shooter, such as *Call of Duty* [4], a software copilot could take control of the actions that the user might struggle to operate, similarly to how **P1**$_P$ does in human cooperation. The software copilot could maneuver the camera orientation, as it requires fine control of input intensity [G1.1] and must be used simultaneously with the movement stick [G1.2], as well as reloading, which must be executed quickly during fast-paced close-combat sequences [G1.3].

5.1.2 Provide Information. In human cooperation, the copilot assists the pilot by also providing information throughout the game. Partial automation can be used to replicate such support as well (Section 4.3.4 and Section 4.3.5). Microsoft is currently investigating a similar type of support as part of their *Copilot* conversational assistant [62], which allows players to explicitly ask for suggestions about the game they are playing. Beyond this form of interaction, based on **providing suggestions on-demand [G2.1]**, our findings reveal that, in some cases, the human copilot may provide information without an explicit request from the pilot. This can also be implemented in partial automation, by detecting when the user needs support and **proactively providing assistance [G2.2]**. Such support can be useful to players with various needs, including individuals with cognitive impairments or beginners. However, these **suggestions should be limited to what is needed to progress when the user is stuck, without spoiling the game [G2.3]** for the pilot (Section 4.6.1). A special type of suggestion, particularly relevant for accessibility, is **signaling important game elements [G2.4]** that the player may not have perceived (Section 4.3.4). This kind of support is reminiscent of Navi[7], the fairy companion of the protagonist in *The Legend of Zelda: Ocarina of Time* [69], who alerts the player of dangers and elements of interest in the game. This type of support could also be used to signal relevant sound cues to Deaf or Hard of Hearing (DHH) people, as recently researched for real-world sound cues [43].

For **P2**$_P$, who is blind and plays action adventure games like *God of War* [81], the software copilot could signal [G2.2] the presence of interactable elements [G2.4], such as levers. However, the copilot should provide such suggestions only if it detects that the user is stuck [G2.3] or the user explicitly requests it [G2.1].

5.1.3 During Game Setup. Human copilots often provide support during game setup, which can involve various hardware and software accommodations required to play a given game (Section 4.3.1). This type of support may be needed even if the game itself can be played independently. Clearly, partial automation cannot support the physical setup, which will still require human support. Instead, it can provide support during software setup, particularly in deciding how to assign commands to the pilot and the copilot. Indeed, while some participants expect the possibility to **decide control assignment explicitly [G3.1]**, others also suggested that partial automation could **propose how to subdivide the controls [G3.2]** (Section 4.4.1). For instance, an experienced gamer like **P2**$_P$ could explicitly personalize the action assignment [G3.1]. Instead, for a pilot with little knowledge of the game, the system could propose a suitable pre-defined configuration [G3.2] based on the pilot's own disability and previous gaming sessions.

Another type of support provided by human copilots is for navigating the game menu, which is a common accessibility problem reported by the participants (Section 4.3.2). A partial automation system could **make the game menu more accessible [G3.3]** by taking control of the game and acting as an actuator that executes the user's explicit instructions (Section 4.4.4). This would solve the menu navigation problem highlighted by **P11**$_{CE}$ in *Forza Horizon* [76] (Section 4.3.1), for example, by allowing the user to verbally command the system to scroll to the desired options and activate them.

5.2 Promoting Autonomy and Sociality

We highlight the positive impact of partial automation on the pilot's autonomy (Section 5.2.1) and discuss the effects of human cooperation and partial automation on the sociality, inclusivity, and engagement of players (Section 5.2.2).

[7] Navi, originally "Fairy Navigation System", provides guidance during the game.

Table 3: Main design guidelines and specific recommendations for shared control

Supporting Different Forms of Assistance	
G1. Delegate control of some game actions	G1.1 Actions requiring fine control of input intensity
	G1.2 Actions requiring concurrent multi-dimensional controls
	G1.3 Actions involved in fast-paced control sequences
G2. Provide information	G2.1 Allow on-demand requests for information
	G2.2 Proactively provide assistance
	G2.3 Provide hints without spoiling the fun of the player
	G2.4 Signal elements of interest in the game
G3. During game setup	G3.1 Facilities for explicit and personalized control assignment
	G3.2 Propose command split between pilot and copilot
	G3.3 Menu navigation support
Promoting Autonomy and Sociality	
G4. Promote autonomy	G4.1 Support for different platforms and games and be tuned on them
	G4.2 Consistent level of support throughout the game
G5. Promote sociality	G5.1 Also support human cooperation (possibly remotely)
	G5.2 Enable assistance in multiplayer video games
Scaffolding Collaboration Between the Pilot and the Copilot	
G6. Policies for subdivision of the controls	G6.1 Full personalization
	G6.2 Agency maximization considering inputs the user can control
	G6.3 Clarity of actions separation: preferably strict and by macro-functionality
G7. Copilot’s operational autonomy	G7.1 Allow on-demand requests by the player
	G7.2 Allow autonomous interventions by the copilot
	G7.3 Avoid doing too much and too well, adapt to the player’s changing abilities
	G7.4 Limit assistance in multiplayer to a baseline or based on player’s abilities
Facilitating Communication and Intent Understanding During the Gameplay	
G8. Communication from the pilot	G8.1 Natural language requests (also using shorthand terms)
	G8.2 Non-verbal commands for people with speech impairment
G9. Communication from the copilot	G9.1 Verbal messages to signal/suggest (also using shorthand terms)
	G9.2 Non-verbal feedback through screen, sounds, light, vibrations
G10. Mutual intent understanding	G10.1 Pilot’s intent: gameplay analysis, external sensing, and active learning
	G10.2 Copilot intent: feedback through visual, auditory, haptic cues

5.2.1 Promote Autonomy. Participants unanimously highlighted one limitation of human cooperation: it requires the availability of a copilot. An additional logistical constraint is that the pilot and the copilot need to be in the same place at the same time (Section 4.2.2). Remote human cooperation removes the constraint of being in the same place, but it still requires the availability of a human copilot and tools that are not widely known and are difficult to set up. The copilot also needs to have adequate gaming abilities and knowledge of the game, both factors which influence the resulting gaming experience (Section 4.6.1). Furthermore, at times, pilots felt that they might ask too much of copilots, and some completely gave up playing to avoid burdening others (Section 4.2.1). These findings mirror broader challenges in assistive contexts. Dependence on helpers’ availability [82] and expertise [90], and the resulting feeling of being a burden are social and emotional implications of human assistance that can restrict users’ autonomy and reduce their sense of agency [5, 67, 82]. Another concern is the potential lack of engagement of the copilot in the game, since the agency is predominantly left to the pilot (Section 4.2.3). While this specific concern may appear unfounded, as the copilots we interviewed were eager to assist their pilots, it still discourages players with disabilities from asking for assistance. All these barriers collectively impede players with disabilities from playing the games they want, when they want, and in the way they want (Section 4.2.1).

For these reasons, participants were excited about the idea of a software copilot, echoing prior research on automated assistance in domains beyond gaming, which highlights the need to reduce reliance on human helpers while preserving the user’s sense of agency [40, 78]. In order to provide widespread and effective assistance, partial automation should **support different platforms and games [G4.1]**. Recent works [83] attest the feasibility of such a solution, contrary to **P1**$_P$’s doubts (Section 4.6.1).

Finally, partial automation should be tuned to **provide the same level of support, and therefore to require the same effort from the player, throughout a game [G4.2]** (Section 4.3.3). For example, **P2**$_P$ reported issues with adapting to different human copilots, with some being more proficient than others (Section 4.6.2). In partial automation, a single software copilot, trained to provide the same level of support at all times, would result in a more consistent and enjoyable user experience across the different levels of a game.

5.2.2 Promote Sociality. Participants reported one major limitation of the idea of substituting human cooperation with partial automation: the resulting loss of live social engagement (Section 4.1.2). This is particularly true for those who organize social moments to play together and bond. Indeed, beyond providing game accessibility, human cooperation also gives the opportunity to play together with friends and family. Thus, participants considered partial automation not as a replacement for human cooperation, but as another option that would allow them more freedom to play in different ways.

This suggests that human cooperation should not be merely considered as an “accessibility hack" or a temporary workaround. Following the concepts of “activity as the ultimate particular" [96] and “design through use" [34], human cooperation can be framed as an authentic form of mixed-ability collaborative play in its own right. It is a practice characterized by mutual adaptation during gameplay among the pilot and the human copilot, from which novel game strategies and collaboration dynamics can emerge [91]. Thus, the results of our research suggest that, despite its limitations, human cooperation should be preserved rather than fully replaced by partial automation. For this reason, shared control systems should **allow users to choose between human cooperation or partial automation [G5.1]**, depending on users' preferences and the availability of a copilot. For example, this would allow **P3**$_P$ to continue playing together with **P4**$_C$ and also play the same games independently when **P4**$_C$ is not available.

The potential limitations of partial automation in terms of social engagement could be partially mitigated by also **enabling assistance in multiplayer video games [G5.2]**, which in turn would have a positive impact on social inclusion. For instance, a team of players using partial automation to play together in multiplayer – each controlling a different character – would be possible, as suggested by **P11**$_{CE}$ and **P12**$_{CE}$ (Section 4.1.2).

5.3 Scaffolding Collaboration Between the Pilot and the Copilot

We discuss how controls should be divided between the pilot and the copilot (Section 5.3.1) and when partial automation should intervene in the game (Section 5.3.2).

5.3.1 Policies for Subdivision of the Controls. A central aspect of shared control is determining how to divide the actions between the pilot and the copilot. Since the subdivision of controls should be defined according to users' preferences and capabilities (Section 4.4.2), it must be **fully customizable [G6.1]**.

As already mentioned in Section 5.1.3, the system should also be able to propose how to subdivide the controls automatically. To this end, we discuss the policies that should guide these configuration proposals. First, the subdivision should consider how many and which inputs the pilot can control. Then, among all the game actions, the ones that provide the most sense of agency should be preferentially assigned to the inputs that the pilot can control, leaving the rest to the copilot. In general, the actions that **maximize the sense of agency [G6.2]** are the ones that are more important for the gameplay or more frequently used. Second, the subdivision of controls should take into account the **clarity of the subdivision [G6.3]**, avoiding sources of automation confusion [20]. Due to this, strict separation should be preferred (Section 4.4.1). Indeed, if both the pilot and the copilot control the same action, it becomes unclear who is responsible for it, and it is harder for the player to understand the consequences of their own actions. For the same reason, the actions belonging to the same macro-functionality should also have the same controller (Section 4.4.2). For example, in a first-person shooter such as *Call of Duty* [4], movement and shooting could be assigned to the pilot, as they are tightly coupled, interdependent **[G6.3]**, and also maximize the pilot's sense of agency [G6.2], while secondary actions like reloading and switching weapons could be delegated to the copilot.

5.3.2 Copilot's Operational Autonomy. Participants were concerned that partial automation interventions could compromise the pilot's sense of control. Thus, they frequently voiced the need to limit the assistance solely to situations of necessity. Interestingly, participants noted that human copilots that are not sufficiently autonomous can only act as actuators, thus spoiling the gaming experience (Section 4.4.4). In contrast, if the software copilot is too skilled to play the game, it can reduce the pilot's sense of agency. Thus, in some cases, they suggested limiting the software agent to only respond to **pilot's explicit commands [G7.1]**. This is suggested, for example, for puzzle or logic games, where the software agent solving the problem can spoil the game for the player (Section 4.4.4). In this case, the pilot could ask the copilot to move a puzzle piece to a specific location, and the copilot would then perform the sequence of game actions needed to achieve this. Instead, for fast-paced games, it could be unfeasible for the pilot to provide specific instructions, and thus the copilot should be able to **intervene autonomously [G7.2]** (Section 5.1.1). For example, if the software agent is responsible for controlling the orientation in a first-person shooter, it is impossible for the pilot to provide precise real-time commands (*e.g.*, through voice) on how to adjust the orientation. It should also be possible to combine explicit actions for which the copilot solely acts as an actuator with other actions in which the copilot is more autonomous. For example, in a first-person shooter, the user can give explicit commands to change the weapon, while the software agent can autonomously aim at enemies.

Additionally, the software **copilot should not be too good, leaving space for the player's sense of agency, and should adapt to the player's changing abilities [G7.3]** (Section 4.3.3), similarly to how **P11**$_{CE}$ and **P12**$_{CE}$ behave when assisting their pilots in human cooperation (Section 4.3.5). Autonomous interventions should also be **limited in multiplayer to avoid benefiting the user unfairly [G7.4]** (Section 4.1.3). To achieve this, the support should be set to a baseline level of abilities (Section 4.3.3). For example, the software copilot should be trained to play on samples from average players [G7.3] and not be tuned to achieve the best results, possibly resulting in being too strong or behaving unnaturally. Alternatively, the software copilot could be configured to **match the player's skills [G7.3, G7.4]** (Section 4.1.3), by replicating the player's reaction times, accuracy, and game style.

5.4 Facilitating Communication and Intent Understanding During the Gameplay

In human cooperation, the communication between the pilot and the copilot is fundamental. Similarly, partial automation should also provide facilities to allow communication from the pilot (Section 5.4.1), feedback from the software copilot (Section 5.4.2), and foster mutual intent understanding (Section 5.4.3).

5.4.1 Communication From the Pilot. To support explicit requests from the pilot to the software copilot, partial automation needs to implement appropriate communication mechanisms. The main use cases for such requests are for the pilot to request some actions to the copilot (Section 4.3.3), such as to change the weapon, or to ask the copilot for suggestions on how to proceed in the game (Section 4.3.5). Participants were interested in making such requests **verbally using natural language [G8.1]** (Section 4.5.1), possibly also teaching personalized shorthand commands to the system for quicker and more comfortable interaction. For example, a player could use a quick personalized command such as "cover" to have the copilot move behind the nearest obstacle in a shooter video game, without having to explain what to do explicitly each time.

Some participants also highlighted the need to **communicate non-verbally [G8.2]**, for example, in the case of players with speech impairments (such as **P7**$_{PC}$) or DHH users. In particular, DHH players might benefit from alternative input methods that do not rely on speech, with visual confirmations of their commands being received and executed. To achieve this, various alternative communication channels could be used, such as controller buttons not used for playing, head [57] or gaze [71] gestures, brain-computer interface [68], or non-verbal mouth sounds [7].

5.4.2 Communication From the Copilot. Software copilot should also **communicate with the pilot, possibly in natural language, to signal elements of interest [G9.1]** in the game (Section 4.3.4) and provide suggestions (Section 4.3.5). The communication should take into account user preferences, such as the user-defined shorthand terms. For example, the software copilot could verbally guide **P2**$_{P}$ when playing *God of War* [81], using quick, synthetic commands like "clear left" or "swoop" (Section 4.5.1).

Besides verbal communication, **non-verbal options should be possible as well [G9.2]**, which may be particularly useful for DHH users. For example, the copilot could display text messages on the screen. Beyond text, symbolic communication can also be conveyed through icons, non-verbal sounds, or haptic feedback, and should be configurable based on user preferences. For example, the copilot could highlight elements of interest on the screen through a visual marker instead of verbally communicating their position. Such awareness cues during gameplay are beneficial to the trust and reliance of the pilot towards the software copilot [22].

5.4.3 Mutual Intent Understanding. Besides responding to explicit requests, the software copilot should also attempt to **infer the user's intent to provide more meaningful automated support [G10.1]**. Such an inference should be made by observing the game and the pilot's actions. Possibly, it could also consider additional external cues, like the pilot's gaze and voice. For example, in a game such as Minecraft [85], the copilot could automatically select the most suitable tool (*e.g.*, pickaxe, shovel, hoe) based on the block the player is interacting with. However, understanding the user's intent can be challenging, in particular when the game does not have a clear goal (Section 4.5.2). To address this issue, participants suggest that the copilot ask the pilot for confirmation on whether their intent is interpreted correctly and learns from the pilot's answer. This can be considered a form of active learning [66]. Of course, the request for confirmation should not interrupt the game's flow.

The system should also support the pilot in understanding the copilot's actions, so that the pilot can factor the copilot's actions into the decision-making process. To this end, **the copilot should provide operational feedback on the actions it performs [G10.2]** (Section 4.3.4), possibly as visual, haptic, or audio cues. For instance, while aiming at an enemy in a first-person shooter game, the software copilot could communicate its intent by displaying a crosshair on the target enemy, so that the pilot can be ready to shoot.

5.5 Generalizability of the Proposed Design Guidelines

The insights from our study and the resulting design guidelines (Table 3) have implications that extend beyond specific video games (Section 5.5.1) and can also apply to different contexts (Section 5.5.2).

5.5.1 Across Video Games. While prior partial-automation systems are often game-specific (Section 2.3.2), our research identifies fundamental forms of support and collaboration applicable across game genres, thereby providing a basis for more generalizable solutions. For instance, the identified forms of copilot support may manifest differently in action, puzzle, or exploration games. In action games, automation could assist with high-frequency motor tasks, such as aiming or camera control; in puzzle games, support may focus on hints to preserve agency; in exploration games, navigation and environmental awareness become central.

Consequently, our design guidelines (Table 3) are formulated to be game-agnostic, focusing on fundamental interaction principles – such as communication mechanisms, control subdivision criteria, and intent understanding – that can be instantiated differently depending on the specific game context. These guidelines could inform the development of both specialized and general-purpose partial automation systems. The first approach would involve designing AI copilots tailored to specific games, enabling fine-grained adaptation to each game's mechanics and accessibility requirements. Alternatively, a single copilot could be trained to provide assistance across multiple games, in a similar way to the SIMA project [83], which introduces a generalist AI capable of autonomously playing various 3D games. Our work contributes to this vision by outlining how such agents should structure their collaboration with human players, defining the communication and control-sharing strategies needed to ensure accessibility and player agency.

5.5.2 To Different Domains. Many proposed design guidelines are generalizable to other domains where humans and software agents collaborate in shared control. For example, the curse of dimensionality challenge (Section 5.1.1) also affects assisted vehicle driving [51], assistive mobility aids (*e.g.* smart wheelchairs [49] or guide robots [47]), and robotic teleoperation [10, 29, 39], including high-stakes contexts like surgical robotics [88]. Consequently, guidelines on delegating control of actions [**G1**] – in particular concurrent multidimensional controls [**G1.2**] and fast-paced control sequences [**G1.3**] – are directly applicable in these domains. Similarly, the guidelines on control subdivision [**G6**] reflect general usability principles: users (acting as pilots) should retain control over actions that maximize their sense of agency [**G6.2**] [99], while maintaining clear separation of actions (**G6.3**) to avoid confusion [70].

Furthermore, the copilot's role in providing information [**G2**] extends naturally to training and educational software [2, 28, 79] that guide learners through explanations, feedback, and adaptive support, and productivity and creative tools (*e.g.*, coding, writing, design) that integrate AI copilots [35, 38] to assist users during content creation by suggesting possible continuations, alternatives, or improvements. The ability to provide on-demand suggestions [**G2.1**], offer proactive assistance when users are stuck [**G2.2**] without directly providing answers [**G2.3**], and highlight relevant interface elements [**G2.4**] represents central functions of these tools.

Similarly, the themes of communication [**G8**, **G9**] and mutual intent understanding [**G10**] are universal in human-AI collaboration. Participants' demand for tunable operational autonomy [**G7**], flexible communication channels (*e.g.*, personalized verbal commands, [**G8.1**]), and clear feedback from the AI [**G10.2**] are essential for building trust and keeping the user engaged in the decision-making.

However, we note that applying our guidelines to other domains requires addressing context-specific aspects: for example, communication modalities that work for gaming may not work elsewhere, and the balance between autonomy and control [**G6.2**] should also account for domain-specific safety requirements and regulatory constraints absent in gaming. For instance, autonomous interventions by a software copilot [G7.2] may pose regulatory issues in domains such as surgical assistance [88]. Additionally, some of our guidelines are intrinsically tied to the gaming context and cannot be generalized to other domains. For example, the guideline to provide hints without spoiling the fun [**G2.3**], the need for the copilot to avoid being too skilled [**G7.3**], and the concern around balancing assistance to prevent cheating in multiplayer video games [**G7.4**] do not translate to safety-critical or productivity-oriented domains, where optimal AI performance and efficiency may be prioritized over preserving the satisfaction of problem-solving.

5.6 Limitations of the Study

In our study, we employed a convenience sampling method to recruit gamers and experts with experience in human cooperation for video game accessibility. This approach allowed us to involve 14 representative participants, which is in line with other accessibility studies [55]. However, we acknowledge that the limited number of participants and the adopted sampling approach introduced some limitations in our study. First, most of our participants have mobility impairments, with a notable exception of **P2**$_P$ who has a visual impairment. Instead, we did not have participants with hearing or cognitive impairments. Second, all our participants come from Western countries, which may introduce a cultural bias to our results. Third, our study did not include participants who were underage or elderly. These issues limit the generalizability of our findings, requiring future investigations to confirm our results.

In addition, our findings are based on participants' recount of their experiences. While all participants had lived experience with human cooperation gaming, none had previously used a partial automation system. Thus, their opinions on partial automation are all hypothetical. A possible extension of this work could therefore be a comparative study of human cooperation and partial automation, with the goal of confirming our results. Moreover, the prompt introducing the partial automation concept during the study was intentionally grounded in participants' existing human cooperation practices (Section 3.2). This methodological choice enabled us to investigate how current human cooperation dynamics could inform the design of partial automation systems. However, we acknowledge that grounding the discussion in existing practices may have constrained participants' feedback, leading them to view partial automation as a substitute for human cooperation rather than exploring it as an entirely novel interaction paradigm. Future research could address this by adopting alternative speculative prompts and engaging users without prior experience in human cooperation, thereby eliciting design requirements for partial automation systems that extend beyond existing human cooperation models.

6 Conclusions and Future Work

This work investigates current practices of shared control in video game accessibility through interviews and focus groups with players with disabilities, their copilots, and accessibility specialists. Our findings indicate that human cooperation is a crucial means of enabling access to games that would otherwise remain unplayable, while also offering opportunities for social interaction and inclusion. At the same time, reliance on human copilots introduces significant limitations, particularly in terms of autonomy, availability, and engagement. Participants expressed a strong interest in partial automation as a promising extension of shared control. Such systems could increase independence, reduce logistical barriers, and broaden access. Some concerns were also raised, in particular about preserving player agency and maintaining the social value of cooperative play. Accordingly, this study contributes to an understanding of how shared control is used in practice and how players perceive the transition toward automated copilots. On this basis, we outline design guidelines to enable autonomous gaming while preserving the pilot's sense of agency.

Several directions for future research emerge from our findings. First, the investigation could be extended to a broader and more diverse population of participants, including individuals with different types of disabilities, cultural backgrounds, experience with human cooperation, and age groups. Such diversity would provide a more comprehensive picture of shared control practices and the potential generalizability of partial automation solutions. Second, observational studies of real-world gaming sessions are needed to complement self-reported accounts and refine our understanding of how human cooperation unfolds in practice. Such studies could reveal additional design considerations and inform the development of automation features that more closely replicate or enhance human support. Finally, the design guidelines identified in this work pave the way for prototyping and experimentally evaluating partial automation systems. Future studies could involve user testing of such prototypes, ideally in comparison with a human cooperation baseline, to assess their effectiveness, usability, and impact on player autonomy, agency, and enjoyment.

References

[1] 8BitDo. n.d.. *8BitDo Lite SE*. https://www.8bitdo.com/lite-se

[2] Khan Academy. n.d.. *Khanmigo*. https://www.khanmigo.ai

[3] ACM. 2021. *ACM Publications Policy on Research Involving Human Participants and Subjects*. https://www.acm.org/publications/policies/research-involving-human-participants-and-subjects

[4] Activision. 2003. Call of Duty. Video game series. https://www.callofduty.com

[5] Dustin Adams, Lourdes Morales, and Sri Kurniawan. 2013. A qualitative study to support a blind photography mobile application. In *Proceedings of the 6th International Conference on PErvasive Technologies Related to Assistive Environments* (Rhodes, Greece) *(PETRA '13)*. ACM, Article 25, 8 pages. doi:10.1145/2504335.2504360

[6] Juan Aguado-Delgado, José-Maria Gutierrez-Martinez, José R Hilera, Luis De-Marcos, and Salvador Otón. 2020. Accessibility in video games: a systematic review. *Universal Access in the Information Society* 19, 1 (2020), 169–193.

[7] Dragan Ahmetovic, Daniele Riboli, Cristian Bernareggi, and Sergio Mascetti. 2021. RePlay: Touchscreen Interaction Substitution Method for Accessible Gaming. In *Proceedings of the 23rd International Conference on Mobile Human-Computer Interaction*. ACM, Article 27, 12 pages. doi:10.1145/3447526.3472044

[8] Cara Aitchison. 2003. From leisure and disability to disability leisure: developing data, definitions and discourses. *Disability & Society* 18, 7 (2003), 955–969. doi:10.1080/0968759032000127353

[9] Game Developers Association. n.d.. *Game Accessibility Guidelines*. https://gameaccessibilityguidelines.com

[10] P.G. Backes and K.S. Tso. 1990. UMI: an interactive supervisory and shared control system for telerobotics. In *Proceedings., IEEE International Conference on Robotics and Automation*, Vol. 2. IEEE, Cincinnati, OH, USA, 1096–1101. doi:10.1109/ROBOT.1990.126141

[11] Cynthia L. Bennett, Jane E, Martez E. Mott, Edward Cutrell, and Meredith Ringel Morris. 2018. How Teens with Visual Impairments Take, Edit, and Share Photos on Social Media. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. ACM, 1–12. doi:10.1145/3173574.3173650

[12] Kevin Bierre, Jonathan Chetwynd, Barrie Ellis, D Michelle Hinn, Stephanie Ludi, and Thomas Westin. 2005. Game not over: Accessibility issues in video games. In *Proc. of the 3rd International Conference on Universal Access in Human-Computer Interaction*. Lawrence Erlbaum Associates (CRC Press), Mahwah, NJ, USA, 22–27.

[13] Kevin Bierre, Michelle Hinn, Teresa Martin, Michael McIntosh, Tess Snider, Katie Stone, and Thomas Westin. 2004. *Accessibility in Games: Motivations and Approaches*. White paper. International Game Developers Association Game Accessibility Special Interest Group. White paper.

[14] Jeffrey P. Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C. Miller, Robin Miller, Aubrey Tatarowicz, Brandyn White, Samual White, and Tom Yeh. 2010. VizWiz: nearly real-time answers to visual questions. In *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology* (New York, New York, USA) *(UIST '10)*. ACM, 333–342. doi:10.1145/1866029.1866080

[15] Mark Brown and Sky LaRell Anderson. 2021. Designing for disability: Evaluating the state of accessibility design in video games. *Games and Culture* 16, 6 (2021), 702–718.

[16] Michelle A Brown and I Scott MacKenzie. 2013. Evaluating video game controller usability as related to user hand size. In *Proceedings of the International Conference on Multimedia and Human Computer Interaction*. ASET, Toronto, Ontario, Canada, 1–9.

[17] ByoWave. n.d.. *Proteus Controller*. https://byowave.com/products/proteus-controller

[18] Jared E. Cechanowicz, Carl Gutwin, Scott Bateman, Regan Mandryk, and Ian Stavness. 2014. Improving player balancing in racing games. In *Proceedings of the First ACM SIGCHI Annual Symposium on Computer-Human Interaction in Play* (Toronto, Ontario, Canada) *(CHI PLAY '14)*. ACM, 47–56. doi:10.1145/2658537.2658701

[19] Gabriele Cimolino, Sussan Askari, and T.C. Nicholas Graham. 2021. The Role of Partial Automation in Increasing the Accessibility of Digital Games. *Proc. ACM Hum.-Comput. Interact.* 5, CHI PLAY, Article 266 (Oct. 2021), 30 pages. doi:10.1145/3474693

[20] Gabriele Cimolino, Renee (Xinyu) Chen, Carl Gutwin, and T.C. Nicholas Graham. 2023. 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* (Hamburg, Germany) *(CHI '23)*. ACM, Article 569, 19 pages. doi:10.1145/3544548.3581116

[21] Gabriele Cimolino and T.C. Nicholas Graham. 2022. Two Heads Are Better Than One: A Dimension Space for Unifying Human and Artificial Intelligence in Shared Control. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. ACM, Article 8, 21 pages. doi:10.1145/3491102.3517610

[22] Gabriele Cimolino, Carl Gutwin, and T.C. Graham. 2022. Impact of Awareness Cues on Trust in Human-AI Shared Control. In *TRAIT: Workshop on Trust and Reliance in AI–Human Teams, at CHI 2022*. ACM.

[23] ConsoleTuner. n.d.. *Titan Two*. https://www.consoletuner.com/products/titan-two

[24] Flex Controller. n.d.. *Flex Controller*. https://www.flex-controller.com

[25] Mat Dalgleish. 2024. *Who Can Play? Rethinking Video Game Controllers and Accessibility*. Springer, 43–71. doi:10.1007/978-3-031-34374-2_3

[26] Guillermo del Toro. 2013. Pacific Rim. Warner Bros. Pictures and Legendary Pictures. (Motion picture).

[27] Paul Dietz and Darren Leigh. 2001. DiamondTouch: a multi-user touch technology. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology* (Orlando, Florida) *(UIST '01)*. ACM, 219–226. doi:10.1145/502348.502389

[28] James Dominic, Charles Ritter, and Paige Rodeghero. 2020. Onboarding Bot for Newcomers to Software Engineering. In *Proceedings of the International Conference on Software and System Processes* (Seoul, Republic of Korea) *(ICSSP '20)*. ACM, 91–94. doi:10.1145/3379177.3388901

[29] Anca D Dragan and Siddhartha S Srinivasa. 2013. A policy-blending formalism for shared control. *The International Journal of Robotics Research* 32, 7 (2013), 790–805. doi:10.1177/0278364913490324

[30] Nintendo EAD. 2017. Mario Kart 8 Deluxe. Video game, released on Nintendo Switch. https://store.nintendo.it/mario-kart

[31] Sony Interactive Entertainment. 2023. *PS5 Access Controller*. https://www.playstation.com/accessories/access-controller

[32] Be My Eyes. n.d.. *Be My Eyes*. https://www.bemyeyes.com

[33] Daniela Feth, Binh An Tran, Raphaela Groten, Angelika Peer, and Martin Buss. 2009. *Shared-Control Paradigms in Multi-Operator-Single-Robot Teleoperation*. Springer, Berlin, Heidelberg, 53–62. doi:10.1007/978-3-642-10403-9_6

[34] William W. Gaver, Jacob Beaver, and Steve Benford. 2003. Ambiguity as a resource for design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA) *(CHI '03)*. ACM, 233–240. doi:10.1145/642611.642653

[35] GitHub. n.d.. *GitHub Copilot*. https://github.com/features/copilot

[36] David Gonçalves, Manuel Piçarra, Pedro Pais, João Guerreiro, and André Rodrigues. 2023. "My Zelda Cane": Strategies Used by Blind Players to Play Visual-Centric Digital Games. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. ACM, Article 289, 15 pages. doi:10.1145/3544548.3580702

[37] David Gonçalves, André Rodrigues, Mike L. Richardson, Alexandra A. de Sousa, Michael J. Proulx, and Tiago Guerreiro. 2021. Exploring Asymmetric Roles in Mixed-Ability Gaming. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. ACM, Article 114, 14 pages. doi:10.1145/3411764.3445494

[38] Google. n.d.. *Gemini in Google Docs*. https://support.google.com/docs/answer/14206696

[39] Deepak Gopinath, Siddarth Jain, and Brenna D Argall. 2016. Human-in-the-loop optimization of shared autonomy in assistive robotics. *IEEE robotics and automation letters* 2, 1 (2016), 247–254. doi:10.1109/LRA.2016.2593928

[40] Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. 2018. VizWiz Grand Challenge: Answering Visual Questions from Blind People. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, Salt Lake City, UT, USA, 3608–3617. doi:10.1109/CVPR.2018.00380

[41] Kotaro Hara, Shiri Azenkot, Megan Campbell, Cynthia L. Bennett, Vicki Le, Sean Pannella, Robert Moore, Kelly Minckler, Rochelle H. Ng, and Jon E. Froehlich. 2015. Improving Public Transit Accessibility for Blind Riders by Crowdsourcing Bus Stop Landmark Locations with Google Street View: An Extended Analysis. *ACM Trans. Access. Comput.* 6, 2, Article 5 (March 2015), 23 pages. doi:10.1145/2717513

[42] Bastian Ilsø Hougaard, Ingeborg Goll Rossau, Jedrzej Jacek Czapla, Mózes Adorján Mikó, Rasmus Bugge Skammelsen, Hendrik Knoche, and Mads Jochumsen. 2021. Who Willed It? Decreasing Frustration by Manipulating Perceived Control through Fabricated Input for Stroke Rehabilitation BCI Games. *Proc. ACM Hum.-Comput. Interact.* 5, CHI PLAY, Article 235 (Oct. 2021), 19 pages. doi:10.1145/3474662

[43] Jeremy Zhengqi Huang, Hriday Chhabria, and Dhruv Jain. 2023. "Not There Yet": Feasibility and Challenges of Mobile Sound Recognition to Support Deaf and Hard-of-Hearing People. In *Proceedings of the 25th International ACM SIGACCESS Conference on Computers and Accessibility* (New York, NY, USA) *(ASSETS '23)*. ACM, Article 15, 14 pages. doi:10.1145/3597638.3608431

[44] Susan Hwang, Adrian L. Jessup Schneider, Daniel Clarke, Alexander Macintosh, Lauren Switzer, Darcy Fehlings, and T.C. Nicholas Graham. 2017. 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* (Edinburgh, United Kingdom) *(DIS '17)*. ACM, 699–710. doi:10.1145/3064663.3064664

[45] JoyToKey. n.d.. *JoyToKey*. https://joytokey.net

[46] Rie Kamikubo, Hernisa Kacorri, and Chieko Asakawa. 2024. "We are at the mercy of others' opinion": Supporting Blind People in Recreational Window Shopping with AI-infused Technology. In *Proceedings of the 21st International Web for All*

Conference (Singapore, Singapore) *(W4A '24)*. ACM, 45–58. doi:10.1145/3677846.3677860

[47] Rie Kamikubo, Seita Kayukawa, Yuka Kaniwa, Allan Wang, Hernisa Kacorri, Hironobu Takagi, and Chieko Asakawa. 2025. Beyond Omakase: Designing Shared Control for Navigation Robots with Blind People. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '25)*. ACM, Article 671, 17 pages. doi:10.1145/3706598.3714112

[48] Eamonn Keogh and Abdullah Mueen. 2017. *Curse of Dimensionality*. Springer, Boston, MA, 314–315. doi:10.1007/978-1-4899-7687-1_192

[49] Ayse Kucukyilmaz and Yiannis Demiris. 2018. Learning Shared Control by Demonstration for Personalized Wheelchair Assistance. *IEEE Transactions on Haptics* 11, 3 (2018), 431–442. doi:10.1109/TOH.2018.2804911

[50] Jonathan Lazar, Jinjuan Heidi Feng, and Harry Hochheiser. 2017. *Research methods in human-computer interaction*. Morgan Kaufmann.

[51] Mingjun Li, Haotian Cao, Xiaolin Song, Yanjun Huang, Jianqiang Wang, and Zhi Huang. 2018. Shared control driver assistance system based on driving intention and situation assessment. *IEEE Transactions on Industrial Informatics* 14, 11 (2018), 4982–4994. doi:10.1109/TII.2018.2865105

[52] Guanhong Liu, Xianghua Ding, Chun Yu, Lan Gao, Xingyu Chi, and Yuanchun Shi. 2019. "I Bought This for Me to Look More Ordinary": A Study of Blind People Doing Online Shopping. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. ACM, 1–11. doi:10.1145/3290605.3300602

[53] Anna Loparev, Walter S. Lasecki, Kyle I. Murray, and Jeffrey P. Bigham. 2014. Introducing shared character control to existing video games. In *International Conference on Foundations of Digital Games*. doi:10.1184/R1/6470180.v1

[54] William Lu. 2008. Evolution of video game controllers. *Roseville: Prima Publishing* 2 (2008).

[55] Kelly Mack, Emma McDonnell, Dhruv Jain, Lucy Lu Wang, Jon E. Froehlich, and Leah Findlater. 2021. What Do We Mean by "Accessibility Research"? A Literature Survey of Accessibility Papers in CHI and ASSETS from 1994 to 2019. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. ACM, Article 371, 18 pages. doi:10.1145/3411764.3445412

[56] Dario Maggiorini, Marco Granato, Laura Anna Ripamonti, Matteo Marras, and Davide Gadia. 2019. Evolution of Game Controllers: Toward the Support of Gamers with Physical Disabilities. In *Computer-Human Interaction Research and Applications*. Springer, Cham, 66–89. doi:10.1007/978-3-030-32965-5_4

[57] Matteo Manzoni, Dragan Ahmetovic, and Sergio Mascetti. 2024. Personalized Facial Gesture Recognition for Accessible Mobile Gaming. In *Computers Helping People with Special Needs*. Springer, Cham, 120–127. doi:10.1007/978-3-031-62846-7_15

[58] Jesse J Martinez, Jon E. Froehlich, and James Fogarty. 2024. Playing on Hard Mode: Accessibility, Difficulty and Joy in Video Game Adoption for Gamers with Disabilities. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. ACM, Article 524, 17 pages. doi:10.1145/3613904.3642804

[59] Lucas Medeiros and Flavio Coutinho. 2015. Developing an Accessible One-Switch Game for Motor Impaired Players. *Proceedings of SBGames* (2015), 236–239.

[60] Microsoft. 2017. *Xbox Controller Assist (formerly Xbox Copilot)*. https://support.xbox.com/help/account-profile/accessibility/copilot

[61] Microsoft. 2023. *Xbox Celebrates Commitment to Accessibility on Global Accessibility Awareness Day 2023*. https://news.xbox.com/en-us/2023/05/18/xbox-celebrates-commitment-to-accessibility

[62] Microsoft. 2025. *Gaming Copilot*. https://news.xbox.com/2025/08/06/gaming-copilot-beta-begins-rolling-out-to-xbox-insiders-on-game-bar-today

[63] Microsoft. n.d.. *Xbox Accessibility Guidelines V3.2*. https://learn.microsoft.com/gaming/accessibility/guidelines

[64] Microsoft. n.d.. *Xbox Adaptive Controller*. https://www.xbox.com/accessories/controllers/xbox-adaptive-controller

[65] Microsoft. n.d.. *Xbox Adaptive Joystick*. https://www.xbox.com/accessories/controllers/xbox-adaptive-joystick

[66] Robert Munro Monarch. 2021. *Human-in-the-Loop Machine Learning: Active learning and annotation for human-centered AI*. Manning, Shelter Island, NY, USA.

[67] Meredith Ringel Morris, Annuska Zolyomi, Catherine Yao, Sina Bahram, Jeffrey P. Bigham, and Shaun K. Kane. 2016. "With most of it being pictures now, I rarely use it": Understanding Twitter's Evolving Accessibility to Blind Users. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. ACM, 5506–5516. doi:10.1145/2858036.2858116

[68] Tutan Nama and Debasis Samanta. 2025. QC Speller: User Interface Design of a Hands-Free Touch-Free Speller with Brain Electroencephalogram Sensory Rhythm. *ACM Trans. Access. Comput.* 17, 4, Article 20 (Dec. 2025), 36 pages. doi:10.1145/3705733

[69] Nintendo. 1998. The Legend of Zelda: Ocarina of Time. Video game, originally released on Nintendo 64, and later on Nintendo GameCube. https://www.nintendo.co.jp/n01/n64/software/zelda

[70] Raja Parasuraman and Victor Riley. 1997. Humans and Automation: Use, Misuse, Disuse, Abuse. *Human Factors* 39, 2 (1997), 230–253. doi:10.1518/001872097778543886

[71] Youn Soo Park and Roberto Manduchi. 2024. A Functional Usability Analysis of Appearance-Based Gaze Tracking for Accessibility. In *Proceedings of the 2024 Symposium on Eye Tracking Research and Applications* (Glasgow, United Kingdom) *(ETRA '24)*. ACM, Article 54, 7 pages. doi:10.1145/3649902.3656363

[72] PlatinumGames. 2014. Bayonetta. Video game, originally released on Xbox 360, PlayStation 3, and later on Nintendo Wii U, Nintendo Switch, Xbox One, PlayStation 4, and PC.

[73] PlatinumGames. 2014. Bayonetta 2. Video game, originally released on Nintendo Wii U and later on Nintendo Switch.

[74] PlayAbility. n.d.. *PlayAbility*. https://www.playability.gg

[75] Playdead. 2016. Inside. Video game. https://playdead.com/games/inside

[76] Playground Games. 2012. Forza Horizon. Video game series. https://forza.net/horizon

[77] Priori Data. 2025. *How Many Gamers Are There in 2025? Latest Stats*. https://prioridata.com/number-of-gamers

[78] Kyle Rector, Keith Salmon, Dan Thornton, Neel Joshi, and Meredith Ringel Morris. 2017. Eyes-Free Art: Exploring Proxemic Audio Interfaces For Blind and Low Vision Art Engagement. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 3, Article 93 (Sept. 2017), 21 pages. doi:10.1145/3130958

[79] David Roldán-Álvarez and Francisco J. Mesa. 2024. Intelligent Deep-Learning Tutoring System to Assist Instructors in Programming Courses. *IEEE Transactions on Education* 67, 1 (2024), 153–161. doi:10.1109/TE.2023.3331055

[80] Marco C Rozendaal, Bram AL Braat, and Stephan AG Wensveen. 2010. Exploring sociality and engagement in play through game-control distribution. *Ai & Society* 25 (2010), 193–201. doi:10.1007/s00146-009-0245-y

[81] Santa Monica Studio. 2005. God of War. Video game series. https://www.playstation.com/god-of-war

[82] Kristen Shinohara and Jacob O. Wobbrock. 2011. In the shadow of misperception: assistive technology use and social interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI '11)*. ACM, 705–714. doi:10.1145/1978942.1979044

[83] SIMA Team, Maria Abi Raad, Arun Ahuja, Catarina Barros, Frederic Besse, Andrew Bolt, Adrian Bolton, Bethanie Brownfield, Gavin Buttimore, Max Cant, Sarah Chakera, Stephanie C. Y. Chan, Jeff Clune, Adrian Collister, Vikki Copeman, Alex Cullum, Ishita Dasgupta, Dario de Cesare, Julia Di Trapani, Yani Donchev, Emma Dunleavy, Martin Engelcke, Ryan Faulkner, Frankie Garcia, Charles Gbadamosi, Zhitao Gong, Lucy Gonzales, Kshitij Gupta, Karol Gregor, Arne Olav Hallingstad, Tim Harley, Sam Haves, Felix Hill, Ed Hirst, Drew A. Hudson, Jony Hudson, Steph Hughes-Fitt, Danilo J. Rezende, Mimi Jasarevic, Laura Kampis, Rosemary Ke, Thomas Keck, Junkyung Kim, Oscar Knagg, Kavya Kopparapu, Rory Lawton, Andrew Lampinen, Shane Legg, Alexander Lerchner, Marjorie Limont, Yulan Liu, Maria Loks-Thompson, Joseph Marino, Kathryn Martin Cussons, Loic Matthey, Siobhan Mcloughlin, Piermaria Mendolicchio, Hamza Merzic, Anna Mitenkova, Alexandre Moufarek, Valeria Oliveira, Yanko Oliveira, Hannah Openshaw, Renke Pan, Aneesh Pappu, Alex Platonov, Ollie Purkiss, David Reichert, John Reid, Pierre Harvey Richemond, Tyson Roberts, Giles Ruscoe, Jaume Sanchez Elias, Tasha Sandars, Daniel P. Sawyer, Tim Scholtes, Guy Simmons, Daniel Slater, Hubert Soyer, Heiko Strathmann, Peter Stys, Allison C. Tam, Denis Teplyashin, Tayfun Terzi, Davide Vercelli, Bojan Vujatovic, Marcus Wainwright, Jane X. Wang, Zhengdong Wang, Daan Wierstra, Duncan Williams, Nathaniel Wong, Sarah York, and Nick Young. 2024. Scaling Instructable Agents Across Many Simulated Worlds. arXiv:2404.10179 [cs.RO] https://arxiv.org/abs/2404.10179

[84] SpecialEffect. n.d.. *SpecialEffect DevKit*. https://specialeffectdevkit.info

[85] Mojang Studios. 2011. Minecraft. Video game.

[86] Warhorse Studios. 2018. Kingdom Come: Deliverance. Video game, released on Microsoft Windows, PlayStation 4, and Xbox One. https://www.deepsilver.com/games/kingdom-come-deliverance

[87] Philipp Sykownik, Katharina Emmerich, and Maic Masuch. 2018. Exploring Patterns of Shared Control in Digital Multiplayer Games. In *Advances in Computer Entertainment Technology*. Springer, Cham, 847–867. doi:10.1007/978-3-319-76270-8_57

[88] Russell Taylor, Pat Jensen, Louis Whitcomb, Aaron Barnes, Rajesh Kumar, Dan Stoianovici, Puneet Gupta, ZhengXian Wang, Eugene Dejuan, and Louis Kavoussi. 1999. A steady-hand robotic system for microsurgical augmentation. *The International Journal of Robotics Research* 18, 12 (1999), 1201–1210.

[89] Gareth Terry, Nikki Hayfield, Victoria Clarke, Virginia Braun, et al. 2017. Thematic analysis. *The SAGE handbook of qualitative research in psychology* 2, 17-37 (2017), 25.

[90] Anja Thieme, Cynthia L. Bennett, Cecily Morrison, Edward Cutrell, and Alex S. Taylor. 2018. "I can do everything but see!" – How People with Vision Impairments Negotiate their Abilities in Social Contexts. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. ACM, 1–14. doi:10.1145/3173574.3173777

[91] Laia Turmo Vidal and Jared Duval. 2024. Ambiguity as a Resource to Design for a Plurality of Bodies. In *Proceedings of the Halfway to the Future Symposium* (Santa Cruz, CA, USA) *(HttF '24)*. ACM, Article 21, 9 pages. doi:10.1145/3686169.3686176

[92] Turn 10 Studios. 2023. *Forza Motorsport Accessibility Support*. https://support.forzamotorsport.net/hc/articles/20964254277267-Forza-Motorsport-Accessibility-Support

[93] Ultraleap. n.d.. *VR Accessibility Recommendations*. https://docs.ultraleap.com/ultralab/vr-gaming-accessibility-recommendations.html

[94] Rodrigo Vicencio-Moreira, Regan L. Mandryk, Carl Gutwin, and Scott Bateman. 2014. 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* (Toronto, Ontario, Canada) *(CHI '14)*. ACM, 937–946. doi:10.1145/2556288.2557308

[95] VoiceAttack. n.d.. *VoiceAttack*. https://voiceattack.com

[96] Annika Waern and Jon Back. 2017. Activity as the Ultimate Particular of Interaction Design. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. ACM, 3390–3402. doi:10.1145/3025453.3025990

[97] Dakuo Wang, Elizabeth Churchill, Pattie Maes, Xiangmin Fan, Ben Shneiderman, Yuanchun Shi, and Qianying Wang. 2020. From Human-Human Collaboration to Human-AI Collaboration: Designing AI Systems That Can Work Together with People. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI EA '20)*. ACM, 1–6. doi:10.1145/3334480.3381069

[98] Bei Yuan, Eelke Folmer, and Frederick C Harris Jr. 2011. Game accessibility: a survey. *Universal Access in the information Society* 10, 1 (2011), 81–100. doi:10.1007/s10209-010-0189-5

[99] Debora Zanatto, Mark Chattington, and Jan Noyes. 2021. Sense of Agency in Human-Machine Interaction. In *Advances in Neuroergonomics and Cognitive Engineering*. Springer, Cham, 353–360. doi:10.1007/978-3-030-80285-1_41