

# How Well Can 3D Accessibility Guidelines Support XR Development? An Interview Study with XR Practitioners in Industry

Daniel Killough  
dkillough@wisc.edu  
University of  
Wisconsin-Madison  
United States

Tiger F. Ji  
reatreify@gmail.com  
University of  
Wisconsin-Madison  
United States

Kexin Zhang  
kzhang284@wisc.edu  
University of  
Wisconsin-Madison  
United States

Yixin Hu  
yixin.hu@wisc.edu  
University of  
Wisconsin-Madison  
United States

Yu Huang  
yu.huang@vanderbilt.edu  
Vanderbilt University  
Nashville, Tennessee  
United States

Ruofei Du  
me@durofei.com  
Google XR Labs  
San Francisco, California  
United States

Yuhang Zhao  
yuhang.zhao@cs.wisc.edu  
University of  
Wisconsin-Madison  
United States

## Abstract

While accessibility (a11y) guidelines exist for 3D games and virtual worlds, their applicability to extended reality (XR)'s unique interaction paradigms (e.g., spatial tracking, kinesthetic interactions) remains unexplored. XR practitioners need practical guidance to successfully implement a11y guidelines under real-world constraints. We present the first evaluation of existing 3D a11y guidelines applied to XR development through semi-structured interviews with 25 XR practitioners across diverse organization contexts. We assessed 20 commonly-agreed a11y guidelines from six major resources across visual, motor, cognitive, speech, and hearing domains, comparing practitioners' development practices against guideline applicability to XR. Our investigation reveals that guidelines can be highly effective when designed as transformation catalysts rather than compliance checklists, but fundamental mismatches exist between existing 3D guidelines and XR requirements, creating both implementation barriers and design gaps. This work provides foundational insights towards developing a11y guidelines and support tools that address XR's distinct characteristics.

## CCS Concepts

- Human-centered computing → Accessibility; Mixed / augmented reality; Virtual reality.

## Keywords

Accessibility, A11y, AR, VR, augmented and virtual reality, extended reality, XR, developer interviews, guidelines

## ACM Reference Format:

Daniel Killough, Tiger F. Ji, Kexin Zhang, Yixin Hu, Yu Huang, Ruofei Du, and Yuhang Zhao. 2026. How Well Can 3D Accessibility Guidelines Support XR Development? An Interview Study with XR Practitioners in Industry. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing*



Please use nonacm option or ACM Engage class to enable CC licenses  
This work is licensed under a Creative Commons Attribution 4.0 International License.  
CHI '26, April 13–17, 2026, Barcelona, Spain  
© 2026 Copyright held by the owner/author(s).  
ACM ISBN 979-8-4007-2278-3/2026/04  
<https://doi.org/10.1145/3772318.3790520>

*Systems (CHI '26), April 13–17, 2026, Barcelona, Spain.* ACM, New York, NY, USA, 15 pages. <https://doi.org/10.1145/3772318.3790520>

## 1 Introduction

Extended reality (XR) technologies create immersive experiences that fundamentally differ from traditional 2D interfaces or 3D virtual worlds displayed on flat screens. XR's unique interaction paradigms, including kinesthetic input, novel spatial interactions, and embodied sense of presence, introduce unique accessibility (a11y) challenges that existing guidelines fail to address and fundamentally alter how a11y must be conceptualized and implemented [30, 61].

While recently organizations have started developing a11y guidelines for 3D virtual worlds (e.g., *Virtual Environments Accessibility Guidelines* [57], *W3CXR* [60], *Game Accessibility Guidelines* [22]), these frameworks were developed primarily from virtual experiences on 2D screens rather than the diverse XR ecosystem spanning virtual reality (VR), augmented reality (AR), and mixed reality (MR). More critically, no published work has systematically explored how XR practitioners—the main force who put guidelines into practice—experience, interpret, and adapt these a11y guidelines to XR development realities. This evaluation gap hinders widespread a11y adoption, as guidelines that practitioners cannot understand, implement, or see as relevant to their work will fail to improve a11y outcomes regardless of their theoretical soundness [16].

While recent work by Wang et al. [61] has identified organizational and motivational barriers to XR a11y implementation, practitioners actively seeking to create inclusive experiences lack adequate support methods and tools [16, 66]. The gap between practitioner intent and available resources motivates our investigation into what enables XR a11y implementation. Understanding what support methods (e.g., guidelines, toolkits) work and why others fail is critical for enabling practitioners to implement a11y within real-world development constraints.

We evaluated 20 a11y guidelines from established virtual world a11y resources with 25 XR practitioners, uncovering specific technical incompatibilities and implementation barriers unique to XR. Our investigation employs a semi-structured interview approach

where guidelines serve as evaluation “techniques” that practitioners apply to their own development practices. This methodology creates a dual evaluation approach where practitioners simultaneously assess their development practices against guidelines while reflecting on the guidelines themselves for clarity, actionability, and applicability. Beyond guideline evaluation, we explored practitioners’ perspectives on additional a11y support tools (e.g., toolkits, 3rd party plugins, automated a11y checking) to understand what support methods best enable efficient a11y implementation within typical XR development constraints. Through this lens we explore the human experiences behind guideline interpretation and implementation, uncovering how professional communities construct meaning around a11y requirements and navigate competing values within XR development cultures.

Our research addresses three questions through qualitative investigation:

- RQ1.** What technical solutions are XR practitioners currently employing for a11y? What technical barriers prevent direct guideline implementation in XR?
- RQ2.** How do XR practitioners interpret existing 3D a11y guidelines when applied to immersive contexts?
- RQ3.** What adaptations would make guidelines actionable for XR development? What additional support methods would enable efficient a11y integration within development constraints?

Our findings revealed: (1) fundamental tensions between immersive design values and a11y implementation that create unique professional challenges; (2) community practices for interpreting and adapting 3D guidelines to XR contexts; (3) technical and methodological incompatibilities between existing 3D guidelines and XR interaction paradigms; and (4) practitioners’ recommendations for a11y support methods to integrate with existing XR development workflows (e.g., improvements for more actionable guidelines).

This work contributes the first exploration of how XR practitioners experience a11y guidelines in their professional contexts, providing empirical evidence for improving guideline design, developing better a11y support tools, and understanding the social and cultural factors that influence accessible XR development.

## 2 Related Work

Our work examines how XR practitioners evaluate existing a11y guidelines, revealing systematic gaps in their applicability to immersive contexts and the social factors influencing implementation decisions. We contextualize our work within existing a11y guidelines for 3D virtual worlds and XR as well as prior literature on guideline implementation in traditional development contexts.

### 2.1 A11y Guidelines for Virtual Worlds and XR

People with disabilities (PWD) face diverse challenges in XR, including kinesthetic interactions for wheelchair users [28], unpredictability of VR content increasing concerns for people with photosensitivity [55], and navigating social environments [15]. Despite extensive assistive technology development for XR (e.g., [13, 31, 45, 49, 52, 65, 67]), most research prototypes remain disconnected from industry practice, with only a small number of 3D games having incorporated a11y features beyond basic settings like colorblind

options (e.g., [21, 62, 64]), highlighting the barriers of large-scale a11y integration in industry.

To support a11y integration in mainstream technologies, researchers and industry developers have created a11y guidelines to serve as foundational resources directing practitioners in implementing inclusive design features. Standard a11y guidelines have been constructed for web [11, 36, 46, 59] and mobile devices [58], and have been used in various studies evaluating mobile platforms [6, 20, 63]. Dedicated a11y guidelines for 3D virtual worlds have also been constructed [1, 22, 42] alongside research efforts for more specific use cases (e.g., movement-based VR games [39, 47] or expressing cultural heritage [14, 54]). For example, the *Game Accessibility Guidelines* (GAG) [22] organize recommendations by disability type (motor, visual, cognitive, speech & hearing) and provide implementation examples ranging from high-level design principles to concrete technical implementations. Similarly, the *Xbox Accessibility Guidelines* (Xbox) [42] and *Accessible Player Experiences* (APX) [1] offer comprehensive frameworks for making 3D games more inclusive. These guidelines emerged primarily from desktop-based virtual worlds where 3D environments are viewed on flat screens with traditional input devices. These 3D a11y guidelines have potential to be applied to XR accessibility as they address challenges absent from solely 2D interface standards (e.g., spatial audio design; depth perception requirements; motion-based interactions; 3D information presentation). However, they may also face fundamental limitations in XR contexts due to XR’s unique interaction paradigms, such as embodied first-person experience and spatial, kinesthetic interactions [28]. It is therefore unclear whether and how the existing 3D a11y guidelines may transfer to the emerging XR applications, for example, which recommendations may transfer effectively and which require fundamental reconceptualization for XR’s unique interaction paradigms.

Beyond 3D a11y guidelines, researchers [30, 40, 44, 51] and organizations [3, 29, 37, 50] have started compiling a11y guidelines specifically for XR experiences. Heilemann et al. [30] compiled VR-relevant guidelines from existing game a11y resources, while organizations like Meta [50] and Magic Leap [37] created device-specific recommendations for their platforms. Despite these ongoing efforts, current XR a11y guidelines are in their infancy, without rigorous validation or broad agreement. Moreover, these frameworks **mostly focus on accommodating an end-user rather than technical guidance to developers in how to implement these guidelines; they also remain largely untested with the practitioners responsible for implementing them, creating uncertainty about their practical applicability within real-world XR development contexts.** To our knowledge, no prior research has systematically validated the applicability and issues of existing 3D and XR a11y guidelines, and examined whether XR practitioners can successfully interpret and implement these guidelines when developing for immersive platforms.

### 2.2 Implementing Guidelines

Researchers have identified challenges of incorporating a11y into traditional development [5, 9, 10, 43], revealing substantial variance in a11y development quality across developers and highlighting how professional identity and organizational context influence

implementation decisions [10] and adding PWD-focused user testing [5]. Understanding how practitioners experience a11y guidelines has emerged as critical for effective implementation; established guideline evaluation methodologies demonstrate that practitioners can effectively assess both their practices against guidelines and evaluate guideline quality when provided with structured frameworks [6, 11, 63]. These approaches have successfully validated web and mobile a11y standards while revealing critical insights about guideline design and implementation patterns.

For XR specifically, Wang et al.'s investigation [61] represents the most comprehensive examination of VR practitioners' general a11y perspectives, revealing challenges including hardware limitations, insufficient professional knowledge, and competing development priorities. Additional research has explored a11y challenges in XR environments [17, 48] and broader XR development challenges [2, 35]. While these foundational works establish understanding of practitioner attitudes toward a11y, they focus on general perspectives rather than evaluating specific guidelines. **Our work extends this foundation by focusing specifically on how practitioners encounter, interpret, and evaluate existing a11y guidelines rather than exploring general attitudes and barriers toward a11y.** Beyond interviews about a11y challenges, we asked XR practitioners to systematically examine existing XR-relevant a11y guidelines and discuss their interpretation, potential use, and implementation concerns through shared projects in their preferred work environment.

### 3 Method

We compiled existing 3D a11y guidelines and conducted 25 semi-structured interviews with XR practitioners to evaluate guideline feasibility and applicability to XR development contexts. Practitioners assessed both their development practices against guidelines and evaluated the guidelines themselves for XR applicability, creating a dual evaluation that reveals both technical implementation barriers and guideline design gaps specific to XR contexts.

#### 3.1 Participants

We interviewed 25 practitioners (23 male, 2 female) with rich XR development experience. Twenty-four participants had 2–9 years experience ( $mean = 4.92$ ,  $SD = 1.94$ ); P17 had less than one year of XR development experience but worked on an a11y team at a major XR game engine. Participants represented diverse organizational contexts: freelancers ( $n = 7$ ); startups ( $n = 6$ ); midsize companies (like game engines, universities, and healthcare organizations) ( $n = 6$ ); and big tech ( $n = 6$ ), covering perspectives across different resource and constraint environments that significantly impact a11y implementation approaches [61]. To ensure qualified evaluation of a11y guidelines, we pre-screened participants prioritizing those with a11y development experience and public contributions to XR projects: 20 participants had prior XR a11y experience, with most experienced in visual a11y ( $n = 10$ ), followed by motor ( $n = 7$ ), cognitive ( $n = 5$ ), speech ( $n = 4$ ), and hearing ( $n = 3$ ). Four participants had experience creating tools for XR platforms. Most participants have development experience with Unity ( $n = 23$ ), with additional experience using Unreal Engine ( $n = 11$ ) and various webXR ( $n = 6$ ) or custom/company-specific engines ( $n = 3$ ). We

detail participants' demographic information, including individuals' years of experience and preferred platform(s), in Table 1.

#### 3.2 Apparatus: Guideline Selection

We collected a11y guidelines applicable to XR through a systematic literature search combining “Augmented Reality,” “Virtual Reality,” “Mixed Reality,” “Extended Reality,” and “Game” with “accessibility guidelines” across Google and Google Scholar. While there are still no commonly-agreed, mature XR a11y standards, we collected existing preliminary guidelines distilled by researchers, a11y organizations, and industry. Specifically, we used Google Scholar as our primary search database (as it aggregates content from major academic databases including ACM Digital Library and IEEE Xplore) to collect guidelines derived from research studies (e.g., [30, 60]), and we supplemented the results with a more general Google search to identify practitioner-facing guidelines from industry (e.g., [37, 42, 50]) and a11y organizations (e.g., [1, 22, 32]). We included game guidelines due to significant design paradigm overlap between XR applications and 3D games [30].

Our search yielded 12 resources: [1, 20, 22, 27, 30, 32, 37, 42, 50, 51, 58, 60]. To narrow the scope we focused on relatively mature, comprehensive guideline resources, excluding guidelines that primarily provided conceptual information without specific a11y issues and implementation recommendations (e.g., [51]) or guidelines that were a subset of more comprehensive resources already included in our analysis (e.g., [30, 58]). We narrowed down to six resources, including the *Game Accessibility Guidelines* (GAG) [22], *W3C XR Accessibility User Requirements* (W3CXR) [60], *Meta Quest Accessible VR Design* (Quest) [50], *Accessible Player Experiences* (APX) [1], *Xbox Accessibility Guidelines* (Xbox) [42], and *IGDA GASIG Top Ten* (IGDA) [32].



**Figure 1: Screenshots as shown to participants from the Game Accessibility Guidelines used as an anchor, showing Mot-1: *Allow controls to be remapped / reconfigured* as an example. We provided original wording (left) and visual examples (right) to participants and gave standardized explanations of each guideline via a slides presentation. Retrieved from [23].**

Based on these guideline resources, we sought to select a commonly-agreed subset of guidelines to present to the XR practitioners in our study, ensuring the representativeness of the guidelines and to avoid overloading the participants with too many guidelines.

PID	Gender	Company Size Exp	Years XR Exp: Primary Role(s)	Primary Platform(s)	Prior XR A11y Exp	Evaluated
P1	Male	startup, freelance	5y: designer, writer, developer	Unity	Motor, Cognitive	Cognitive
P2	Female	startup, freelance	6.5y: developer	Unity	Visual, Motor	Visual
P3	Male	big tech	4y: manager, researcher, developer	BabylonJS Playground	Visual; XR tools	S&H
P4	Male	startup, freelance	4y: manager, designer, developer	Unity	Visual, Motor	Cognitive
P5	Male	big tech, freelance	3y: designer, animator, developer	Unity	Speech, Hearing (S&H)	Motor
P6	Male	startup, freelance	2.5y: manager, developer	Glitch	WebXR tools	Visual
P7	Male	midsize	5y: researcher, developer	Unity	Visual, Cognitive	Motor
P8	Male	startup, freelance	9y: designer, developer	Custom Engine; Unity	None	Cognitive
P9	Male	midsize	2y: developer	Glitch	None	Visual
P10	Female	midsize	3y: developer	Unity	None	S&H
P11	Male	startup, freelance	5y: developer	Unreal	None	Motor
P12	Male	startup	3.5y: designer	Unity	Hearing	S&H
P13	Male	startup	2y: developer	Unity	Motor	Motor
P14	Male	midsize, freelance	3y: researcher, developer	PlayCanvas	Visual	S&H
P15	Male	big tech	5.5y: designer, developer	Unity	Visual; XR tools	Visual
P16	Male	midsize, startup	4y: designer, researcher, developer	Unity	Cognitive; XR tools	Cognitive
P17	Male	midsize	<1y: a11y developer	Redacted XR Engine	A11y team at XR Engine	Visual
P18	Male	startup	6y: executive, designer, researcher	Unity	None	S&H
P19	Male	big tech	8y: developer	Unity	Visual, Motor	S&H
P20	Male	startup	7y: team lead, developer	Unity	Visual	Cognitive
P21	Male	big tech, startup	5y: developer	Unity	Motor	Cognitive
P22	Male	startup, freelance	5y: team lead, developer	Unity	Cognitive	Cognitive
P23	Male	big tech, startup	8y: designer, developer	Unity	Visual, Motor, Speech	Visual
P24	Male	big tech, midsize	7y: developer	Unity	Visual	Visual
P25	Male	big tech	5y: team lead, developer	Custom Engine; Unreal	Cognitive, S&H	S&H

**Table 1: Participant Demographic Information, including gender, organizational context(s), years of XR development experience and responsibilities, primary development platforms, years of XR a11y development experience, and guidelines evaluated.**

We compared and cross-referenced the guidelines in the six resources, organized the guidelines by disability types (visual, motor, cognitive, speech/hearing), and selected five most commonly mentioned guidelines across the six resources for each disability group. We merged the guidelines for speech and hearing disabilities as both emerged as highly relevant to communication, and they had a smaller number of guidelines than other disability types. Some resources already merged these guidelines; for example, the GAG noted text chat as benefiting d/Deaf and non-verbal users together [24]. This process yielded 20 guidelines with five in each disability group, which we have aggregated in Table 2.

When presenting the selected guidelines to the XR practitioners, we used GAG as our “anchor” resource as it represented the most sophisticated and comprehensive resource at the time of our study [30], providing elaboration, PWD experience, and best practice examples for each guideline (see Fig. 1). During interviews, we presented participants with the actual GAG guideline wording and visual examples via a slideshow presentation as shown in Fig. 1, and provided links to original source materials for reference, ensuring that participants evaluated the guidelines as published.

### 3.3 Procedure

To evaluate these guidelines with XR practitioners we conducted 1.5–3 hour semi-structured interviews via Zoom (October 2022–July 2023), compensating participants at \$50/hr. Each interview included three phases:

**Background & XR A11y Experience.** Participants first described their demographic information, XR development experience, and typical project workflows. We then explored participants’ a11y knowledge and XR-specific a11y experience. Participants with a11y experience detailed their development processes, motivations, strategies, and testing approaches. Those without XR a11y experience discussed barriers to implementation. Participants finally screen-shared one XR project to demonstrate their development practices.

**Guideline Evaluation.** In this core phase, participants reviewed and evaluated the XR a11y guidelines we collected (§ 3.2). Due to time constraints, each participant evaluated one disability type (visual, motor, cognitive, or speech & hearing) based on their interests and experience. We used a preference-matching approach: participants selected two preferred disability types, and we chose one to ensure balanced coverage across all guideline groups while respecting participant expertise. *After cleaning the data, we resulted in seven participants evaluating visual, four motor, seven cognitive, and seven speech & hearing guidelines. We believe we reached saturation with this count.* After determining the group to evaluate, we

Disabilities	Guideline ID	Guideline Content	References
Motor Impairments	<i>Mot-1:</i> Remap Controls	Allow controls to be remapped / reconfigured. (See Fig. 1)	GAG, APX, IGDA, Quest, W3CXR, Xbox
	<i>Mot-2:</i> Flexible Inputs	Support more than one input device.	GAG, APX, IGDA, W3CXR, Xbox
	<i>Mot-3:</i> Body-Agnostic	Do not rely on motion tracking of specific body types.	GAG, APX, Quest, W3CXR, Xbox
	<i>Mot-4:</i> Simple Controls	Ensure controls are as simple as possible, or provide a simpler alternative.	GAG, APX, IGDA, Quest, W3CXR, Xbox
	<i>Mot-5:</i> Flexible Timing	Do not make precise timing essential to gameplay—offer alternatives, actions that can be carried out while paused, or a skip mechanism.	GAG, APX, IGDA, W3CXR, Xbox
Visual Impairments	<i>Vis-1:</i> Supplement Color	Ensure no essential information is conveyed by a fixed colour alone.	GAG, APX, IGDA, Quest, Xbox
	<i>Vis-2:</i> Resizable UI	Allow interfaces to be resized.	GAG, APX, IGDA, W3CXR, Xbox
	<i>Vis-3:</i> Readable Font Size	Use an easily readable default font size.	GAG, APX, Quest, Xbox
	<i>Vis-4:</i> Audio Description	Provide an audio description track.	GAG, APX, W3CXR, Xbox
	<i>Vis-5:</i> Text Voiceovers	Provide pre-recorded voiceovers for all text, including menus and installers.	GAG, IGDA, Quest, W3CXR, Xbox
Cognitive Impairments	<i>Cog-1:</i> Avoid Flicker	Avoid flickering images and repetitive patterns.	GAG, APX, Quest, W3CXR, Xbox
	<i>Cog-2:</i> Supplement Text	Ensure no essential information (especially instructions) is conveyed by text alone, reinforce with visuals and/or speech.	GAG, APX, IGDA, Quest, W3CXR, Xbox
	<i>Cog-3:</i> Symbol Chat	Use symbol-based chat (smileys, etc.).	GAG, APX, IGDA, W3CXR, Xbox
	<i>Cog-4:</i> Adjustable Speed	Include an option to adjust the game speed.	GAG, APX, IGDA, W3CXR, Xbox
	<i>Cog-5:</i> Hide Distractions	Provide an option to turn off / hide background movement.	GAG, APX, IGDA, W3CXR, Xbox
Speech & Hearing Impairments	<i>SH-1:</i> Visualize Sound	Provide captions or visuals for significant background sounds.	GAG, APX, IGDA, Quest, W3CXR, Xbox
	<i>SH-2:</i> Multimodal Chat	Support text chat as well as voice for multiplayer.	GAG, APX, IGDA, W3CXR, Xbox
	<i>SH-3:</i> Supplement Audio	Ensure no essential information is conveyed by sounds alone.	GAG, APX, IGDA, Quest, W3CXR, Xbox
	<i>SH-4:</i> Separate Volumes	Provide separate volume controls or mutes for effects, speech and background / music.	GAG, APX, IGDA, Quest, W3CXR, Xbox
	<i>SH-5:</i> Subtitle Settings	Allow subtitle/caption presentation to be customised.	GAG, APX, IGDA, Quest, W3CXR

**Table 2: Selected XR A11y Guidelines, grouped by disability type (Motor, Visual, Cognitive, and Speech & Hearing). Guidelines are abbreviated with unique IDs and short reference names for identification under the Guideline ID column. The Guideline Content presents the original wording of the guideline summary from the GAG resource. Original wording for each guideline from different sources are hyperlinked under the References column (note that similar guideline content could be presented differently in different resources).**

followed a structured seven-step evaluation protocol to evaluate each guideline: (1) Researcher presents guideline using a prepared slides presentation as a visual aid, showing guidelines' original wording from GAG, published examples from source (Fig. 1), and links to original resources for reference; (2) Participant provides initial interpretation of the guideline in their own words; (3) Researcher provides standardized explanation if interpretation differs; (4) Participant assesses implementation feasibility within their specific XR development context; (5) Detailed discussion of anticipated technical challenges and implementation barriers; (6) Participant suggests adaptations or improvements to the guideline for better applying to XR contexts; (7) Participant assesses external support needed (tools, resources, expertise) for successful implementation. This protocol enabled systematic comparison across participants while capturing contextual factors affecting guideline applicability.

**Development Support Preferences.** We concluded by exploring participants' perspectives and preferences on a11y integration beyond guidelines (e.g., toolkits, *post hoc* solutions) and resource allocation attitudes.

### 3.4 Analysis

Interviews were recorded with Zoom and transcribed by a professional online service approved by the University of Wisconsin-Madison Institutional Review Board (IRB). Researchers reviewed

each transcript to correct errors. We analyzed transcripts using thematic analysis [12], starting with two researchers open coding the same set of three participants independently, then meeting to calibrate interpretations and resolve disagreements through discussion. Through iterative discussion across multiple coding sessions, researchers ultimately agreed on all codes used for 100% inter-coder reliability. One researcher coded the remaining 22 transcripts, updating the codebook as new codes emerged upon agreement with the research team. Analysis prioritized themes revealing: (1) gaps between XR development practices and guidelines' suggestions, (2) interpretation challenges when applying 3D guidelines to XR and implementation barriers specific to XR contexts, and (3) suggested modifications for XR-appropriate guidelines.

## 4 Findings

From interviews with 25 XR practitioners, we present key findings revealing practitioners' approaches to implementing accessible XR features, their interpretation and evaluation of existing XR-adjacent a11y guidelines, and their requirements for additional XR a11y support methods.

## 4.1 Current XR A11y Solutions and Gaps in Practice

Supporting prior work [61], most participants (17 out of 25) had little to no formal XR training, instead learning through online resources ( $n = 14$ ), on the job ( $n = 8$ ), or from personal projects ( $n = 3$ ). Furthermore, no participant indicated formal a11y training. Despite limited resources and guidance, many XR practitioners have created innovative a11y solutions that could inform future standardization efforts, extending beyond the general implementation approaches documented by prior work. Our participants demonstrated unique technical approaches that reveal both successful patterns and critical gaps in current practice.

**4.1.1 Leveraging Platform Capabilities for A11y.** Some practitioners strategically used their platform's capabilities to implement a11y features. For example, when developing mobile AR applications, instead of using Unity that did not "hook" well with the native screen reader (e.g., VoiceOver for iOS, Talkback for Android) (P3, P17), P3 leveraged "Native" development platforms (e.g., Babylon Native) and added invisible 2D buttons on top of 3D objects in their scene, so that the screen reader can access the alt text of the 2D button to read out the 3D object: *"These invisible buttons, you could cycle through just as you would any other button. And they could kind of highlight with the screen reader focus indicator, ... And it would ... give you an object, or it'll basically tell you the manipulation of the 3D objects that you [can] do."* This approach demonstrates practitioners' adaptability leveraging existing platform capabilities and choosing suitable platforms based on a11y needs.

**4.1.2 Alternative Input Integration.** Practitioners have integrated diverse input mechanisms to accommodate different a11y needs, providing practical solutions to motor guidelines (e.g., *Mot-1: Remap Controls*). Participants reported using controllers like the Xbox Adaptive Controller (P20) or libraries like Rewired [25] (P5) and Unity's XR Input system. P17 worked at an XR game engine company and his team incorporated keyboard navigation into their XR controller system, *"So any game that already had controller support now suddenly had keyboard support,"* enabling alternate control methods for many new games developed on their engine. This solution demonstrates efficient a11y feature scaling but heavily relies on development platforms' support.

**4.1.3 Professional Tensions in A11y Implementation.** Nearly half of our participants ( $n = 12$ ) reported a fundamental tension between creating immersive XR experiences and implementing a11y features, extending findings from prior work on developer attitudes [61]. At least three practitioners (P4, P10, P20) explicitly mentioned prioritizing their game's *feel* over users' a11y needs. For example, P4 hesitated to remove flickering visual effects despite awareness of seizure risks: *"For some app designs, just trying to incorporate certain a11y guidelines is going to be a **cursed problem that you cannot solve**. And just to be aware of those problems at a design stage and conscientiously make that trade off, like—we care enough about developing this app that we are willing to exclude this chunk of potential players."* This decision to exclude PWD reveals how practitioners construct professional identity through conscious choices about whose experiences matter most, pulling against their willingness to implement guidelines.

**4.1.4 Motivating Developer Adoption.** Echoing prior work [19], we found that general best design considerations for XR a11y often correlate with general best practices for XR applications ( $n = 10$ ). For example, practitioners considered their app being one-handable to be a proper design consideration (P2, P24) rather than an a11y feature. As P12 stated: *"[The] best argument for a11y is usability. The fact that like, good a11y isn't just for people with impairments, it also makes the product better."*

P6 recommended framing guidelines as "VR guidelines for everybody" or "How to make [VR] comfortable for everyone" rather than focusing solely on PWD to avoid XR developers "brushing off" guidelines' suggestions. Nearly all participants seek hard data on how many users they would gain by addressing each feature ( $n = 21$ ), the estimated time necessary to implement features ( $n = 19$ ), and approximate metrics of difficulty versus value for each feature. If a11y guidelines are framed as general design best practices, then the number of users becomes *everyone* rather than framing PWD as an "other."

Practitioners indicated that popular apps should act as guideposts and standards for a11y implementation. P4, P21, and P24 specifically wanted guideline implementation examples that reference adoption in existing or popular apps (e.g., Beat Saber's colorblindness options, shown in Appendix B). When users become familiar with an a11y feature in one app, they may expect it in others, making implementation easier to justify and replicate once there's an established point of reference.

## 4.2 How Practitioners Experience and Evaluate A11y Guidelines

Our central contribution lies in understanding how practitioners interpret and evaluate existing 3D a11y guidelines when implementing their XR apps. While practitioners were familiar with *some* existing a11y guidelines, they lacked experience applying them to XR apps. Even known guidelines proved insufficient for XR; for example, P3 and P20 cited Microsoft's MR Accessibility Standards [41] as not "fleshed out," while P3 and P17 noted that the W3C's WCAG [59] only applies to 2D interfaces. P24 observed that XR lacks basic system-wide a11y features like screen magnifiers despite prior research demonstrating their feasibility (e.g., SeeingVR's magnifier tool [67]). Although a11y guidelines exist for traditional platforms and some 3D environments, our findings reveal significant gaps highlighting the need for accessibility guidelines specifically designed for XR's unique interaction paradigms.

**4.2.1 Practitioners' Post-Guideline Solutions.** Our systematic evaluation revealed participants' ability to translate abstract guidelines to concrete technical solutions. All 25 practitioners came up with a11y implementation solutions based on the guidelines that can potentially be integrated in their current projects. For example, after evaluating *SH-1: Visualize Sound*, P3 proposed an a11y-conscious technical architecture that fits into a common 'good design' practice: *"Say if you have a prefab that [manages] audio, like send[s] data to your visual sound indicator. So no matter, as long as you have your prefabs all extend that single singular like script or object or hook or whatever, then it should be fairly simple."* P17 and P18 referenced Fortnite's radial sound indicator [18] as an existing implementation

of *SH-1: Visualize Sound*, while P12 innovated beyond existing solutions, describing a “sound wheel” objective marker that would only display when sounds occurred outside the player’s field of view. For *SH-5: Subtitle Settings*, P18 proposed a minimally-intrusive startup experience as simple as: “*Can you see this text clearly? And if it is, click Yes*,” avoiding the need for users to navigate complex a11y menus.

**4.2.2 Confusion & Ambiguity of the Guidelines.** When evaluating our presented guidelines, practitioners generally reported understanding the three information redundancy guidelines (*Vis-1: Supplement Color*, *Cog-2: Supplement Text*, *SH-3: Supplement Audio*) due to their similarity to established web standards. However, practitioners evaluated other guidelines as too broad or using ambiguous language; for *Cog-5: Hide Distractions*, P16 and P21 struggled to define “non-interactive elements”—does the guideline include atmospheric effects? UI chrome? Background NPCs? Furthermore, *Vis-4: Audio Description* confused P6 and P9 about whether the guideline referred to real-time narration versus pre-recorded tracks; and how it differs from *Vis-5: Text Voiceovers*, interpreting *Vis-4* as inclusive of or an extension of *Vis-5*.

**4.2.3 Applicability & Implementation Challenges in XR Contexts.** Guidelines addressing temporal flexibility (*Mot-5: Flexible Timing* and *Cog-4: Adjustable Speed*) presented unique implementation challenges for XR; while technically possible, participants questioned their applicability to real-time XR experiences: P20’s firefighting simulator requires real-time reactions for training validity; P1, P8, and P13 referenced Beat Saber [7] reducing scores for slower speeds as a fairness mechanism, but we note this implementation penalizes users needing the accommodation. Participants rated these timing-related guidelines as moderate importance but low motivation to implement as a result, struggling with preserving immersion and multiplayer fairness. P8 suggested to “*develop toward your extremes, low and high, and then everything else will kind of modulate in place*”, but this strategy may still prove challenging to practitioners. This tension between a11y and core gameplay mechanics appeared uniquely challenging for XR compared to traditional gaming.

**4.2.4 Priority of Guideline Implementation.** Priority ratings revealed three main tiers based on perceived importance and feasibility:

**Tier 1 (immediate implementation)** included guidelines critical for safety or legal compliance. For example: *Cog-1: Avoid Flicker*, *Vis-1: Supplement Color*, and *SH-3: Supplement Audio*, which participants rated as both highest importance and clearest to currently understand their wording as applied to XR.

**Tier 2 (platform-dependent)** included features participants felt platforms should provide, e.g., *SH-5: Subtitle Settings* and *Vis-2: Resizable UI*. P3 argued the platform should handle subtitle preferences, with developers just responsible for passing data through from the users’ settings. Most guidelines fell in this more moderate range with medium to high importance but similarly moderate difficulty and low motivation to implement.

**Tier 3 (context-specific)** included guidelines with applicability concerns: *Cog-3: Symbol Chat* for non-social apps, or *Cog-4: Adjustable Speed* for real-time experiences. These received lowest motivation scores despite recognized importance, with freelancers

like P1 neutral about implementing the guideline unless their client directly requested the feature.

**4.2.5 Implementation Timing and Responsibilities.** Practitioners’ implementation preferences revealed clear patterns about when and who should integrate a11y into development workflows. We present these findings across all 20 guidelines using a responsibility assignment matrix (RACI) [4] in Table 3 to help streamline complex projects with multiple stakeholders.

**When should a11y be implemented?** Participants overwhelmingly ( $n = 21$ ) felt a11y features should be considered and implemented at the design phase ( $n = 12$ ), project start ( $n = 4$ ), or early development ( $n = 3$ ) when possible, with safety-critical guidelines like *Cog-1: Avoid Flicker* requiring careful planning due to inherent danger (P8). P8 believed it is worth a visual quality sacrifice to improve XR performance for this guideline: “*It’s literally everything. Because if you have a bad experience like this, number one, it could physically hurt somebody*.” Other guidelines, like those for adapting user interfaces for people with visual impairments (*Vis-2: Resizable UI*, *Vis-4: Audio Description*, *Vis-5: Text Voiceovers*), become “prohibitively expensive” late in development (P24): their structural, cross-system impact demands careful design decisions that would necessitate overhauling the system for a clean implementation. P6 emphasized this difficulty: “*Trying to patch [Vis-5] on after the fact sounds like disaster*.”

**Who are responsible for implementing a11y guidelines?** Participants mostly agreed ( $n = 18$ ) that *designers* should be primarily responsible for implementing or enforcing a11y features, particularly guidelines affecting visual presentation and user experience (*Vis-1: Supplement Color*, *Vis-3: Readable Font Size*, *Cog-2: Supplement Text*). In some cases, designers should be less involved in favor of developers (P19 for *SH-2: Multimodal Chat*) or direct feedback from testers (P18 for *SH-3: Supplement Audio*). Fourteen participants agreed that *developers* should be primarily responsible for implementing more technical features, particularly system-level features like *Mot-4: Simple Controls*, *SH-4: Separate Volumes*, and *Mot-2: Flexible Inputs*. We support prior work as practitioners taking on multiple roles often causes a11y to be deprioritized in smaller teams, potentially explaining why startup and midsize practitioners generally showed lower willingness to incorporate a11y features (P4, P7, P9, P20, P23).

### 4.3 Community Visions for Guideline-Informed A11y Support

Including and beyond evaluating guidelines, our findings reveal a critical need for developer-centered support tools that integrate with existing XR development workflows; specifically, a downloadable, free, open-source 3rd party a11y package or set of reusable components emerged as most preferred by practitioners ( $n = 21$ ), but further adoption barriers vary by organizational context.

**4.3.1 Third-Party Tool Adoption Patterns.** Practitioners’ willingness to adopt third-party a11y tools depends on organizational culture and technical constraints. Eleven participants noted using third-party packages in their XR projects, but adoption patterns differ even across these users.

Guideline ID	Designers	Developers	QA / Test	PM / Lead*	Audio Eng	Phase?
<i>Mot-1: Remap Controls</i>	A (Technical)	R	C	I	-	Design
<i>Mot-2: Flexible Inputs</i>	R (Technical)	A	C	I	-	Early Dev
<i>Mot-3: Body-Agnostic</i>	A (Technical)	R	C	I	-	Design
<i>Mot-4: Simple Controls</i>	R (UX)	R	C	I	-	Early Dev
<i>Mot-5: Flexible Timing</i>	R (UX)	R	C	I	-	Early Dev
<i>Vis-1: Supplement Color</i>	R (Visual)	R	C	I	-	Design
<i>Vis-2: Resizable UI</i>	R (UX)	A	C	C	-	Design
<i>Vis-3: Readable Font Size</i>	R (Visual)	A	C	I	-	Early Dev
<i>Vis-4: Audio Description</i>	R (UX)	A	C	C	A	Mid- Dev
<i>Vis-5: Text Voiceovers</i>	R (UX)	A	C	C	A	Early Dev
<i>Cog-1: Avoid Flicker</i>	C (Technical)	A	R	A	-	Design
<i>Cog-2: Supplement Text</i>	R (UX)	A	C	A	C	Early Dev
<i>Cog-3: Symbol Chat</i>	R (UX)	A	R	I	C	Early Dev
<i>Cog-4: Adjustable Speed</i>	R (UX)	R	C	I	-	Mid- Dev
<i>Cog-5: Hide Distractions</i>	R (UX)	A	C	I	-	Post-Launch
<i>SH-1: Visualize Sound</i>	R (UX)	R	C	C	A	Early Dev
<i>SH-2: Multimodal Chat</i>	R (UX)	A	C	I	C	Early Dev
<i>SH-3: Supplement Audio</i>	R (UX)	A	C	C	A	Early Dev
<i>SH-4: Separate Volumes</i>	R (Audio/UX)	R	C	C	A	Mid- Dev
<i>SH-5: Subtitle Settings</i>	R (UX)	A	C	I	C	Post-Launch

**Table 3: Responsibility Assignment Matrix (RACI) with Implementation Phases.** Roles are color-coded based on assignment: **Responsible** (red), **Accountable** (yellow), **Consulted** (blue), and **Informed** (grey). A dash (-) indicates no involvement necessary. Designer types shown in parentheses. The Phase? column shows when guidelines should be implemented: **Design** (app architecture), **Early Dev** (prototyping/MVP), **Mid- Dev** (secondary features/polish), or **Post-Launch** (maintenance).

\*PM/Lead represents the equivalent role for a project manager at each organization's respective size: i.e., Project/Product Manager (big tech/midsized), Technical Lead/Founder (startups), or self-coordinated/client-driven (freelancers).

Startup practitioners exhibit “not-invented-here syndrome” according to P2, with strict feelings against external packages and outside consultants perpetuating “*This sort of philosophy that we know everything.*” Other practitioners with big tech experience may be more likely to pick up 3rd party tools, quickly evaluating their “worth” towards speeding up their development or build time (P15, P19, P23). P23 emphasized a critical pitfall associated with third-party supports: “**You think the tool is giving you the time to implement it.** If a tool takes 25 hours to implement into your application, that's misery. If it takes 25 minutes, that's an actually good tool.” Big tech companies (P3, P15, P25) prefer internal tools or require open-source packages for security compliance. These companies also prefer code-based scene generation over visual editors to mitigate conflicts with version control; as P3 explained, “*When someone changed the prefab, now my day's work is hosed. Because the stuff I did in the GUI with clicking buttons in 12 different places is gone.*” This organizational variation suggests a11y tools must provide multiple integration paths: lightweight, open-source (and appropriately licensed) packages for larger companies, rapid prototyping tools for startups, and extensive customization options for specialized applications.

**4.3.2 Implementation Guidance and Examples.** One of the greatest improvements to guidelines suggested by practitioners is to add

specific a11y feature examples and their implementations ( $n = 20$ ) accompanying each guideline. These requests revealed three categories of desired support:

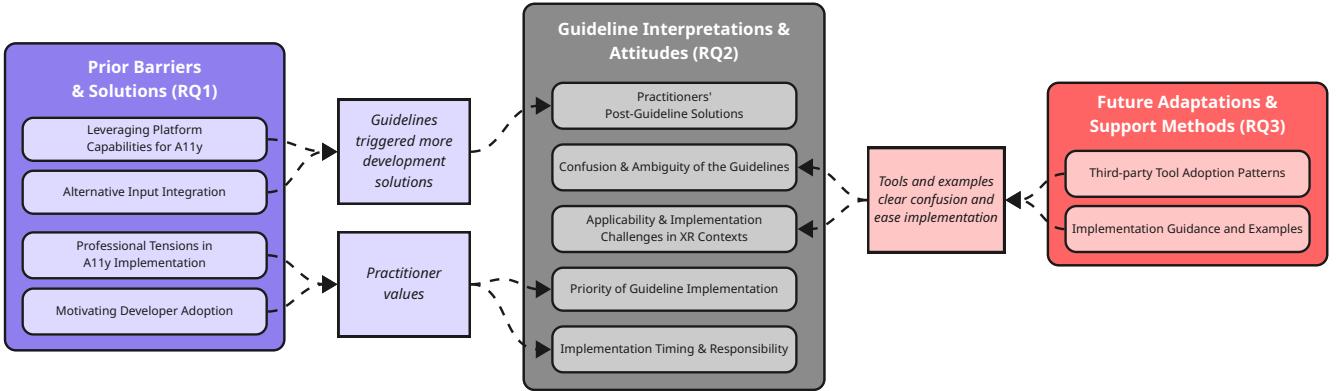
**Comparative examples:** Practitioners wanted both accessible and inaccessible implementations shown side-by-side in an “easily digestible” format (P12, P23), including videos or GIFs (P12, P19, P22) of PWD unable to use an app until the support method is added, potentially making the impact more evocative (P14).

**Code artifacts:** Participants wanted downloadable project materials (P11) including Unity scenes (P3), code snippets with inline documentation (P8), and platform-specific implementations to be easily imported into preferred frameworks: OpenXR ( $n = 7$ ), Unity XR Interaction Toolkit ( $n = 6$ ), and MRTK ( $n = 6$ ) users each requested examples in their platforms.

**Popular app references:** Five participants (P2, P4, P15, P21, P24) want guidelines to reference implementations in successful apps. We showcase the popular VR game Beat Saber's [7] use of colorblind options as an example in Appendix B.

## 5 Discussion

Our evaluation reveals that existing 3D a11y guidelines fail when applied to XR development realities; by examining how practitioners interpret, implement, and adapt existing guidelines, we establish



**Figure 2: Mapping research questions addressed by our findings. RQ1 identified technical solutions, headset limitations, and training gaps. RQ2 revealed ambiguity in applying 3D guidelines to XR contexts and tensions between immersion and a11y. RQ3 showed practitioners need automated checking, concrete examples, and code-first support. Overall, practitioners are willing to implement a11y but need help balancing effort versus impact.**

that inadequate guidance, not just professional culture or organizational constraints, contributes to poor XR a11y adoption. Building on Wang et al. [61]’s work identifying general barriers to XR a11y implementation, our work provides the first practitioner-based assessment of specific guideline applicability to XR.

We document practitioners’ existing a11y solutions and implementation barriers (RQ1), finding that practitioners develop creative workarounds independently, though these remain scattered without structured guidance. When presented with guidelines through structured discussions (RQ2), practitioners demonstrated strong technical ability by proposing concrete implementation strategies and identifying specific technical challenges, revealing both their capacity to translate abstract guidelines into practical approaches and identify the gaps that prevent successful application to XR contexts. However, this evaluation also revealed critical interpretation challenges: practitioners struggled to apply 3D virtual world guidelines to XR’s unique spatial, temporal, and performance constraints, leading to ambiguity about scope and feasibility. Their willingness to adopt guidelines varied by organizational context, reflecting different perceptions of priorities and responsibilities. For support methods, practitioners strongly preferred platform-integrated tools and concrete code examples over static documentation (RQ3), revealing a mismatch between traditional a11y guidance formats and XR development workflows. These findings show that improving XR a11y requires rethinking how a11y knowledge is packaged and delivered to fit XR development practice. We reflect on these tensions in industry contexts in § 5.1 and derive actionable feedback for future support methods in § 5.2.

### 5.1 Professional Tensions Reflect Guidance Gaps

Prior research has documented a11y challenges among software practitioners broadly [10] and the gap between HCI research innovations and industry adoption [16]; our findings extend these insights to reveal how XR’s unique demands create distinct tensions between immersive design values and a11y implementation. Our

evaluation revealed that practitioners’ struggles to balance immersion with a11y stem from inadequate guidance rather than inherent design conflicts (§ 4.1.3). When practitioners express willingness to exclude users with disabilities to preserve their envisioned experience, or resist removing potentially harmful effects despite known safety risks, these decisions reflect a critical gap: existing guidelines fail to demonstrate how a11y and XR design can coexist. Some current 3D guidelines recommend a11y features as additions or modifications to existing designs via separate modes, alternative interfaces, or removed features [1, 22, 42]. This framing positions a11y as potentially compromising the “core” experience, leaving practitioners to believe they must choose between creative vision and inclusive design, or dismiss a11y entirely as not “worth” their effort. However, this perceived trade-off appears to emerge from guidance limitations rather than technical necessity.

**Guidelines Should Demonstrate Universal Benefits.** A11y guidelines should evolve from disability-specific checklists to frameworks demonstrating universal design benefits. As documented in § 4.1.4, our participants recognized that features like alternate input methods and supplemental feedback improve experiences across user populations (even for users who may not identify as PWD, consistent with Microsoft’s Inclusive Design Principles [19]). Guidelines should forefront these universal benefits while maintaining specific disability accommodations, helping practitioners understand a11y as *expanding* rather than constraining design possibilities. This shift requires concrete examples of XR experiences achieving both immersion and broad a11y, but our participants’ struggles to adapt these examples to diverse XR contexts reveal how a lack of guidance forces design choices that unnecessarily pit a11y against immersion.

**Platform-Integrated Support Tools.** Our practitioners strongly ( $n = 21$ ) preferred platform-integrated solutions over manual implementation (§ 4.3.1). Therefore, XR development platforms should provide built-in a11y frameworks that maintain immersive quality. When platforms handle core a11y infrastructures like screen reader

support, subtitle systems, and input remapping, developers can focus on design considerations while leveraging reliable a11y features. In cases where platforms don't implement a11y, recent *post hoc* systems like Killough et al.'s *VRSight* [34] (that generates spatial audio scene descriptions for blind VR/MR users) may provide user-sided compensation without developer integration, demonstrating how technical workarounds can enable a11y without compromising immersive experiences.

**Education and Community Building.** Beyond guidelines and tools, addressing these tensions requires reframing a11y in practitioner culture. Our participants mostly relied on a11y implementations from popular titles (§ 4.3.2), revealing how rarely accessible yet immersive XR experiences are publicly celebrated. The current guidance gap creates false choices between creative vision and inclusive design; filling this gap requires cultural transformation where a11y expertise becomes recognized as fundamental to XR design excellence rather than specialized knowledge only fulfilled for legal compliance.

## 5.2 Informing Future XR A11y Guidelines

Research on translational resources demonstrates that practitioners prefer actionable, prescriptive guidance over abstract principles [16], yet insufficient formal accessibility education often leaves practitioners ill-equipped to implement inclusive designs [53]. Building on our findings, we present design considerations for future XR a11y guidelines, addressing recurring patterns in how practitioners interpret, evaluate, and adopt a11y guidance within real-world XR development constraints. We frame these design insights as considerations rather than definitive recommendations, recognizing that guideline development involves complex trade-offs and that our practitioner sample, while diverse, may not capture all development contexts.

**1. Demonstrate Feasibility via Concrete Examples in XR Contexts.** Practitioners familiar with web a11y encountered confusion applying concepts to XR, highlighting the need for concrete, XR-native examples alongside abstract principles.

*1.1 Provide specific implementation values with citations (§ 4.2.2).*

Specify concrete values (e.g., safe flicker rates in Hz, minimum text sizes, readable distances) with authoritative citations.

*1.2 Include side-by-side visuals showing accessible vs. inaccessible implementations (§ 4.3.2).* Provide comparative examples through GIFs, videos, or screenshots demonstrating compliant and non-compliant designs.

*1.3 Reference successful implementations from popular apps (§ 4.1.4).* Cite examples from widely-recognized apps (e.g., Beat Saber's colorblind options, Appendix B) demonstrating feasibility and commercial viability.

*1.4 Provide downloadable sample scenes demonstrating first-person implementation (§ 4.3.1).* Offer complete examples as Unity and/or Unreal sample projects showing a11y features from the user's perspective (*i.e.*, using an egocentric camera system like Unity's *XR Rig* [56]).

**2. Establish Implementation Timing.** A11y implementation difficulty and cost increase exponentially as projects mature, with post hoc additions often requiring fundamental architectural changes:

*2.1 Implement a11y as early as possible during design phase (§ 4.2.5).* Specify that a11y should be considered during initial project planning and architecture, emphasizing early integration reduces costs while acknowledging post hoc additions remain possible.

*2.2 Build a11y into experience design baselines before asset creation (§ 4.2.5).* Establish a11y systems and frameworks before creating assets to ensure a11y shapes infrastructure rather than being constrained by existing designs.

**3. Address XR-Specific Constraints.** XR's unique technical and interaction constraints (spatial tracking, kinesthetic interactions, headset limitations) require explicit guideline treatment:

*3.1 Never attach UI elements to the camera or head (§ 4.2.4).* Headlocked UI causes motion sickness; specify world-anchored or body-anchored placement strategies.

*3.2 Account for headset fit variability affecting FOV, legibility, and comfort (§ 4.2.4).* Specify adaptive placement strategies and generous safe zones for critical information across diverse users and headset configurations.

*3.3 Support locomotion via multiple methods (§ 4.2.4).* Provide diverse locomotion options including physical movement, controller input, and alternate input methods appropriate for different XR application types.

*3.4 Distinguish AR vs. VR interaction requirements (§ 4.2.4).* Articulate AR vs. VR a11y requirement differences, including AR-specific occlusion handling and outdoor visibility.

*3.5 Test across multiple headset types to account for performance differences (§ 4.2.4).* Specify testing requirements across standalone and tethered headsets to address performance variations and platform-specific capabilities.

**4. Provide Implementation Tools.** Practitioners require concrete tools and infrastructure beyond documentation to efficiently implement a11y features:

*4.1 Offer free, open-source plugins and packages for major engines (§ 4.3.1).* Provide pre-built, production-ready components for Unity and Unreal under open-access licenses to reduce implementation barriers.

*4.2 Include automated a11y checking tools and testing frameworks (§ 4.3.1).* Develop automated a11y checking tools that identify common issues during development.

*4.3 Enable importing platform-level user settings (§ 4.1.1).* Provide mechanisms for XR apps to access users' OS a11y preferences, ensuring consistent experiences across apps.

*4.4 Provide a11y metadata tagging systems (§ 4.2.4).* Develop XR equivalents of alt text to enable developers to semantically describe 3D objects for assistive technologies.

*4.5 Follow OpenXR standards for cross-platform consistency (§ 4.3.1).* Align a11y features with OpenXR specifications to ensure implementations work across diverse hardware platforms.

**5. Frame Motivation and Benefits.** Practitioners need compelling justification for a11y investment given competing priorities and resource constraints:

*5.1 Frame guidelines as general best XR design practices benefiting all users (§ 4.1.4).* Position a11y features as improvements to

overall user experience rather than accommodations for a minority population.

- 5.2 *Quantify expanded market reach through user impact numbers* (§ 4.1.4). Provide concrete statistics on potential users affected by each a11y feature to justify development investment.
- 5.3 *Note that people develop disabilities over time* (§ 4.1.4). Emphasize that a11y benefits extend to aging, temporary impairments, and situational limitations.
- 5.4 *Highlight AAA<sup>1</sup> a11y features as industry-viable* (§ 4.3.2). Reference a11y in commercially successful titles demonstrating compatibility with high-quality, profitable XR experiences.

These considerations emerged from practitioner evaluation of existing guidelines and represent patterns we observed across diverse organizational contexts. However, we acknowledge these considerations require validation through guideline development and deployment studies. Different development contexts may require different approaches, and trade-offs between comprehensiveness and simplicity warrant further investigation.

### 5.3 Limitations and Future Work

Our study focused on evaluating existing guidelines rather than developing new XR-specific guidance. Without systematic practitioner evaluation of current resources, developing new standards would lack empirical grounding about real-world developer needs and constraints. Our evaluation strongly suggests that inadequate guidance, not just professional culture, contributes to systematic exclusion of PWD from XR experiences.

Given that this study is an a11y-focused work, we acknowledge potential recruitment bias toward a11y-supportive practitioners: many participants expressed interest in a11y and many had prior a11y experience; however, even if practitioners are particularly *motivated* to implement a11y, we document significant technical barriers, guideline interpretation challenges, and implementation conflicts that prevent adoption. To enrich our data, future work should expand participant recruitment and consider lightweight methodologies (e.g., questionnaires) to attract broader developers in contributing their experiences.

Our evaluation methodology provides a framework for assessing guideline effectiveness with practitioners, but requires application across broader practitioner populations and diverse XR application contexts. Future research should build on our foundational work to develop, test, and refine new XR-*native* a11y guidelines that address the technical incompatibilities, organizational contexts, and implementation preferences we identified. We call for community contributors to add a11y samples and modules to open-source frameworks like *XR Blocks* [38], contribute to open-source a11y tools like *VRSight* [34] and *SeeingVR* [68], and share their best practices and implementations built on existing game engines.

## 6 Conclusion

Through evaluating existing 3D a11y guidelines with 25 XR practitioners, we demonstrate that current guidelines systemically fail

<sup>1</sup>We use the term “AAA” in this instance to refer both to the games industry classification for high-budget titles produced by major publishers [8] and to WCAG Level AAA, the highest conformance tier [59].

to address XR’s unique technical requirements and development contexts. Practitioners prefer platform-integrated tools over written guidelines, requesting Unity-compatible packages, OpenXR integration, and early design-phase implementation. Our documented interpretation struggles and implementation barriers establish that inadequate guidance, not just professional culture, contributes to poor XR accessibility adoption. Our findings reveal that guidelines addressing timing controls appeared technically feasible but fundamentally inappropriate for real-time XR experiences, while platform-specific challenges like standalone headset performance constraints and spatial audio complexity remained entirely unaddressed by current 3D virtual world guidelines. For guideline creators, our work emphasizes the need for concrete implementation examples with XR-specific technical specifications, clear responsibility assignments across organizational contexts, and frameworks that address XR’s unique constraints— spatial interaction paradigms, real-time performance requirements, and novel input methods— rather than simple adaptations from 2D or desktop 3D contexts. Our evaluation provides essential evidence for developing XR a11y support that addresses real practitioner needs while enabling the inclusive immersive experiences that XR technology promises.

## Acknowledgments

This work was partially supported by the National Science Foundation under Grant No. IIS-2328182. Thank you to everyone who has helped us in publishing this manuscript over the years, including but not limited to: Our participants for their valuable insight; to Justin Feng and Daniel Wang for their assistance aggregating interview data; and to our labmates for their feedback.

## References

- [1] AbleGamers. 2022. *Accessible Player Experiences (APX)*. Retrieved Sep 20, 2022 from <https://accessible.games/accessible-player-experiences/>
- [2] Narges Ashtari, Andrea Bunt, Joanna McGrenere, Michael Nebeling, and Parmit K. Chilana. 2020. Creating Augmented and Virtual Reality Applications: Current Practices, Challenges, and Opportunities. In *CHI '20: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) (*CHI '20*). Association for Computing Machinery, New York, NY, USA, 1–13. doi:10.1145/3313831.3376722
- [3] XR Association. 2020. *XRA'S DEVELOPERS GUIDE, CHAPTER THREE: Accessibility & Inclusive Design in Immersive Experiences* - XR Association — xra.org. Retrieved August 2024 from <https://xra.org/research/xra-developers-guide-accessibility-and-inclusive-design/>
- [4] Atlassian. 2025. *RACI Chart: What is it & How to Use*. <https://www.atlassian.com/work-management/project-management/raci-chart>
- [5] Aleksander Bai, Heidi Mork, and Viktoria Stray. 2017. A Cost-Benefit Analysis of Accessibility Testing in Agile Software Development - Results From a Multiple Case Study. *International Journal On Advances in Software* 10, 1-2 (03 2017), 96–107. <http://urn.nb.no/URN:NBN:no-73911>
- [6] Mars Ballantyne, Archit Jha, Anna Jacobsen, J. Scott Hawker, and Yasmine N. El-Glaly. 2018. Study of Accessibility Guidelines of Mobile Applications (*MUM 2018*). Association for Computing Machinery, New York, NY, USA, 305–315. doi:10.1145/3282894.3282921
- [7] Beat Games. [n. d.]. Beat Saber. <https://www.beatsaber.com>
- [8] Alexander Bernevega and Alex Gelker. 2022. The Industry of Landlords: Exploring the Assetization of the Triple-A Game. *Games and Culture* 17, 1 (2022), 47–69. doi:10.1177/15554120211014151
- [9] Tingting Bi, Xin Xia, David Lo, and Aldeida Aleti. 2021. A First Look at Accessibility Issues in Popular GitHub Projects. In *2021 IEEE International Conference on Software Maintenance and Evolution (ICSM&E)*, 390–401. doi:10.1109/ICSM&E52107.2021.00041
- [10] Tingting Bi, Xin Xia, David Lo, John Grundy, Thomas Zimmermann, and Denae Ford. 2022. Accessibility in Software Practice: A Practitioner’s Perspective. *ACM Trans. Softw. Eng. Methodol.* 31, 4, Article 66 (07 2022), 26 pages. doi:10.1145/

3503508

[11] Giorgio Brajnik. 2009. Validity and Reliability of Web Accessibility Guidelines. In *Proceedings of the 11th International ACM SIGACCESS Conference on Computers and Accessibility* (Pittsburgh, Pennsylvania, USA) (Assets '09). Association for Computing Machinery, New York, NY, USA, 131–138. doi:10.1145/1639642.1639666

[12] Virginia Braun and Victoria Clarke. 2012. *Thematic Analysis*. American Psychological Association.

[13] Inra Choi, Eyal Ofek, Hrvoje Benko, Mike Sinclair, and Christian Holz. 2018. Claw: A Multifunctional Handheld Haptic Controller for Grasping, Touching, and Triggering in Virtual Reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–13. doi:10.1145/3173574.3174228

[14] Hwei Teeng Chong, Chen Kim Lim, Minhaz Farid Ahmed, Kian Lam Tan, and Mazlin Bin Mokhtar. 2021. Virtual Reality Usability and Accessibility for Cultural Heritage Practices: Challenges Mapping and Recommendations. *Electronics* 10, 12 (2021), 1430. doi:10.3390/electronics10121430

[15] Jazmin Collins, Kaylah Myranda Nicholson, Yusuf Khadir, Andrea Stevenson Won, and Shiri Azenkot. 2024. An AI Guide to Enhance Accessibility of Social Virtual Reality for Blind People. In *Proceedings of the 26th International ACM SIGACCESS Conference on Computers and Accessibility*. 1–5.

[16] Lucas Colusso, Cynthia L. Bennett, Gary Hsieh, and Sean A. Munson. 2017. Translational Resources: Reducing the Gap Between Academic Research and HCI Practice. In *Proceedings of the 2017 Conference on Designing Interactive Systems* (Edinburgh, United Kingdom) (DIS '17). Association for Computing Machinery, New York, NY, USA, 957–968. doi:10.1145/3064663.3064667

[17] Chris Creed, Maadh Al-Kalbani, Arthur Theil, Sayan Sarker, and Ian Williams. 2024. Inclusive augmented and virtual reality: A research agenda. *International Journal of Human-Computer Interaction* 40, 20 (2024), 6200–6219.

[18] Mark Delaney. 2021. How To Use Visualize Audio Effects In Fortnite. Retrieved December 2024 from <https://www.gamespot.com/articles/how-to-use-visualize-audio-effects-in-fortnite/1100-6498314/>

[19] Microsoft Inclusive Design. [n. d.]. Inclusive 101 Guidebook. Retrieved August 2024 from <https://inclusive.microsoft.design/tools-and-activities/Inclusive101Guidebook.pdf>

[20] Google Developers. 2021. *Make Apps More Accessible*. Retrieved Sep 20, 2022 from <https://developer.android.com/guide/topics/ui/accessibility/apps>

[21] Naughty Dog. 2020. Accessibility Options for the Last of Us Part II. <https://www.playstation.com/en-us/games/the-last-of-us-part-ii/accessibility/> last accessed 15 Jun 2022.

[22] Barrie Ellis, Gareth Ford-Williams, Lynsey Graham, Dimitris Grammenos, Ian Hamilton, Ed Lee, Jake Manion, and Thomas Westin. 2012. *A Straightforward Reference for Inclusive Game Design*. Retrieved Sep 20, 2022 from <https://gameaccessibilityguidelines.com/>

[23] Barrie Ellis, Gareth Ford-Williams, Lynsey Graham, Dimitris Grammenos, Ian Hamilton, Ed Lee, Jake Manion, and Thomas Westin. 2012. *Allow controls to be remapped / reconfigured*. Retrieved December 2024 from <https://gameaccessibilityguidelines.com/allow-controls-to-be-remapped-reconfigured/>

[24] Barrie Ellis, Gareth Ford-Williams, Lynsey Graham, Dimitris Grammenos, Ian Hamilton, Ed Lee, Jake Manion, and Thomas Westin. 2012. *Support text chat as well as voice for multiplayer*. Retrieved December 2024 from <https://gameaccessibilityguidelines.com/support-text-chat-as-well-as-voice-for-multiplayer/>

[25] Guavaman Enterprises. 2014. Rewired. <https://guavaman.com/projects/rewired/>

[26] Nuša Faric, Henry WW Potts, Adrian Hon, Lee Smith, Katie Newby, Andrew Steptoe, and Abi Fisher. 2019. What players of virtual reality exercise games want: thematic analysis of web-based reviews. *Journal of medical Internet research* 21, 9 (2019), e13833.

[27] Metal Pop Games. 2022. *UI Accessibility Plugin*. Retrieved Sep 20, 2022 from <http://www.metalphopgames.com/assetstore/accessibility/doc/index.html>

[28] Kathrin Gerling, Patrick Dickinson, Kieran Hicks, Liam Mason, Adalberto L Simeone, and Katta Spiel. 2020. Virtual Reality Games for People Using Wheelchairs. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–11. doi:10.1145/3313831.3376265

[29] Nielsen Norman Group. 2021. *10 Usability Heuristics Applied to Virtual Reality*. Retrieved Sep 20, 2022 from <https://www.nngroup.com/articles/usability-heuristics-virtual-reality/>

[30] Fiona Heilemann, Gottfried Zimmermann, and Patrick Münster. 2021. Accessibility Guidelines for VR Games - A Comparison and Synthesis of a Comprehensive Set. *Frontiers in Virtual Reality* 2 (2021), 9 pages. doi:10.3389/fvrir.2021.697504

[31] Wilko Heuten, Daniel Wichmann, and Susanne Boll. 2006. Interactive 3D Sonification for the Exploration of City Maps. In *Proceedings of the 4th Nordic Conference on Human-Computer Interaction: Changing Roles* (Oslo, Norway) (NordiCHI '06). Association for Computing Machinery, New York, NY, USA, 155–164. doi:10.1145/1182475.1182492

[32] IGDA-GASIG. 2019. SIG Guidelines. <http://igda-gasig.org/get-involved/sig-initiatives/resources-for-game-developers/sig-guidelines>

[33] National Eye Institute. 2023. *Types of Color Vision Deficiency*. <https://www.nei.nih.gov/learn-about-eye-health/eye-conditions-and-diseases/color-blindness/types-color-vision-deficiency>

[34] Daniel Killough, Justin Feng, Zheng Xue Ching, Daniel Wang, Rithvik Dyava, Yapeng Tian, and Yuhang Zhao. 2025. VRsight: An AI-Driven Scene Description System to Improve Virtual Reality Accessibility for Blind People. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*. 1–17. doi:10.1145/3746059.3747641

[35] Veronika Krauß, Alexander Boden, Leif Oppermann, and René Reiners. 2021. Current Practices, Challenges, and Design Implications for Collaborative AR/VR Application Development. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) (CHI '21). Association for Computing Machinery, New York, NY, USA, Article 454, 15 pages. doi:10.1145/3411764.3445335

[36] Sri Kurniawan and Panayiotis Zaphiris. 2005. Research-Derived Web Design Guidelines for Older People. In *Proceedings of the 7th International ACM SIGACCESS Conference on Computers and Accessibility* (Baltimore, MD, USA) (Assets '05). Association for Computing Machinery, New York, NY, USA, 129–135. doi:10.1145/1090785.1090810

[37] Magic Leap. 2018. *Accessibility*. Retrieved Sep 20, 2022 from <https://ml1-developer.magicleap.com/en-us/learn/guides/bp-for-accessibility>

[38] David Li, Nels Numan, Xun Qian, Yanhe Chen, Zhongyi Zhou, Evgenii Alekseev, Geonsun Lee, Alex Cooper, Min Xia, Scott Chung, Jeremy Nelson, Xiuixiu Yuan, Jolica Dias, Tim Bettridge, Benjamin Hersh, Michelle Huynh, Konrad Piascik, Ricardo Cabello, David Kim, and Ruofei Du. 2025. XR Blocks: Accelerating Human-centered AI + XR Innovation. In *arxiv*. 9 pages. doi:10.48550/arXiv.2509.25504

[39] Liam Mason, Kathrin Gerling, Patrick Dickinson, Jussi Holopainen, Lisa Jacobs, and Kieran Hicks. 2022. Including the Experiences of Physically Disabled Players in Mainstream Guidelines for Movement-Based Games. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) (CHI '22). Association for Computing Machinery, New York, NY, USA, Article 86, 15 pages. doi:10.1145/3491102.3501867

[40] Owen McGrath. [n. d.]. *XR Accessibility / Universal Design for Learning – udl.berkeley.edu*. Retrieved August 2024 from <https://udl.berkeley.edu/accessibility/xr-accessibility>

[41] Microsoft. 2022. Accessibility—MRTK3. <https://learn.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk3-accessibility/packages/accessibility/overview>

[42] Microsoft. 2023. *Xbox Accessibility Guidelines V3.2*. Retrieved 2024 from <https://learn.microsoft.com/en-us/gaming/accessibility/guidelines>

[43] Darliana Miranda and João Araujo. 2022. Studying Industry Practices of Accessibility Requirements in Agile Development. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing* (Virtual Event) (SAC '22). Association for Computing Machinery, New York, NY, USA, 1309–1317. doi:10.1145/3477314.3507041

[44] Martez Mott, Edward Cutrell, Mar Gonzalez Franco, Christian Holz, Eyal Ofek, Richard Stoakley, and Meredith Ringel Morris. 2019. Accessible by Design: An Opportunity for Virtual Reality. In *2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. 451–454. doi:10.1109/ISMAR-Adjunct.2019.00122

[45] Martez Mott, John Tang, Shaun Kane, Edward Cutrell, and Meredith Ringel Morris. 2020. “I Just Went Into It Assuming That I Wouldn’t Be Able to Have the Full Experience”: Understanding the Accessibility of Virtual Reality for People With Limited Mobility. In *The 22nd International ACM SIGACCESS Conference on Computers and Accessibility* (Virtual Event, Greece) (ASSETS '20). Association for Computing Machinery, New York, NY, USA, Article 43, 13 pages. doi:10.1145/3373625.3416998

[46] Mozilla. 2022. Understanding the Web Content Accessibility Guidelines. [https://developer.mozilla.org/en-US/docs/Web/Accessibility/Understanding\\_WCAG](https://developer.mozilla.org/en-US/docs/Web/Accessibility/Understanding_WCAG) last accessed 15 Jun 2022.

[47] Florian Mueller and Katherine Isbister. 2014. Movement-Based Game Guidelines. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) (CHI '14). Association for Computing Machinery, New York, NY, USA, 2191–2200. doi:10.1145/2556288.2557163

[48] Vinaya Hanuman Naikar, Shwetha Subramanian, and Garrett W Tigwell. 2024. Accessibility feature implementation within free vr experiences. In *Extended abstracts of the CHI conference on human factors in computing systems*. 1–9.

[49] Vishnu Nair, Jay L Karp, Samuel Silverman, Mohar Kalra, Hollis Lehv, Faizan Jamil, and Brian A Smith. 2021. NavStick: Making Video Games Blind-Accessible Via the Ability to Look Around. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. 538–551. doi:10.1145/3472749.3474768

[50] Oculus. 2022. *Designing Accessible VR*. Retrieved Sep 20, 2022 from <https://developer.oculus.com/resources/design-accessible-vr/>

[51] The University of Melbourne. 2022. *Accessibility of Virtual Reality Environments*. Retrieved Sep 20, 2022 from <https://www.unimelb.edu.au/accessibility/virtual-reality>

[52] Lorenzo Picinali, Amandine Afonso, Michel Denis, and Brian FG Katz. 2014. Exploration of architectural spaces by blind people using auditory virtual reality

for the construction of spatial knowledge. *International Journal of Human-Computer Studies* 72, 4 (2014), 393–407.

[53] Cynthia Putnam, Emma J Rose, and Craig M MacDonald. 2023. “It could be better. It could be much worse”: Understanding Accessibility in User Experience Practice with Implications for Industry and Education. *ACM Transactions on Accessible Computing* 16, 1 (2023), 1–25.

[54] Hesmeralda Rojas, Ronald Renteria, Ely Acosta, Humberto Arévalo, and Marisol Pilares. 2020. Application of accessibility guidelines in a virtual museum. In *2020 3rd International Conference of Inclusive Technology and Education (CONTIE)*. 73–79. doi:10.1109/CONTIE51334.2020.00022

[55] Laura South, Caglar Yildirim, Amy Pavel, and Michelle A Borkin. 2024. Barriers to photosensitive accessibility in virtual reality. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–13.

[56] Unity. 2022. XR Interaction Toolkit. <https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@0.2/manual/index.html> last accessed 15 Jun 2022.

[57] University of South Carolina Center for Teaching Excellence. n.d. Virtual Environments Accessibility Guidelines. [https://sc.edu/about/offices\\_and\\_divisions/cte/teaching\\_resources/virtual\\_environments/ve\\_accessibility\\_guidelines/](https://sc.edu/about/offices_and_divisions/cte/teaching_resources/virtual_environments/ve_accessibility_guidelines/). Accessed: 2025-12-04.

[58] W3C. 2015. *Mobile Accessibility: How WCAG 2.0 and Other W3C/WAI Guidelines Apply to Mobile*. Retrieved Sep 20, 2022 from <https://www.w3.org/TR/mobile-accessibility-mapping/#wcag-2.0-and-mobile-content-applications>

[59] W3C. 2018. *Web Content Accessibility Guidelines (WCAG) 2.1*. Retrieved Sep 20, 2022 from <https://www.w3.org/TR/WCAG21/>

[60] W3C. 2020. *XR Accessibility User Requirements*. Retrieved Sep 20, 2022 from <https://www.w3.org/TR/xaur/>

[61] Yi Wang, Xiao Liu, Chetan Arora, John Grundy, and Thuong Hoang. 2025. Understanding VR Accessibility Practices of VR Professionals. In *Proceedings of the 2025 CHI conference on human factors in computing systems*.

[62] WB Games Support. 2024. Accessibility Options. Mortal Kombat Games Support Documentation. <https://mortalkombatgamessupport.wbgames.com/hc/en-us/articles/360023370494-Accessibility-Options>

[63] Zhuxiaona Wei and James A. Landay. 2018. Evaluating Speech-Based Smart Devices Using New Usability Heuristics. *IEEE Pervasive Computing* 17, 2 (2018), 84–96. doi:10.1109/MPRV.2018.022511249

[64] World's Edge studio and Relic Entertainment. 2024. Age IV Accessibility Features. Age of Empires Official Website. <https://www.ageofempires.com/age-iv-accessibility/>

[65] Momona Yamagami, Sasa Junuzovic, Mar Gonzalez-Franco, Eyal Ofek, Edward Cutrell, John Porter, Andrew Wilson, and Martez Mott. 2021. Two-in-One: A Design Space for Mapping Unimanual Input Into Bimanual Interactions in Vr for Users With Limited Movement. *ACM Transactions on Accessible Computing* 15, 3 (2021), 1–25. doi:10.1145/3510463

[66] Kexin Zhang, Jr. Spencer, Edward Glenn Scott, Abijith Manikandan, Andric Li, Ang Li, Yaxing Yao, and Yuhang Zhao. 2025. Inclusive Avatar Guidelines for People with Disabilities: Supporting Disability Representation in Social Virtual Reality. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 560, 26 pages. doi:10.1145/3706598.3714230

[67] Yuhang Zhao, Edward Cutrell, Christian Holz, Meredith Ringel Morris, Eyal Ofek, and Andrew D. Wilson. 2019. SeeingVR: A Set of Tools to Make Virtual Reality More Accessible to People With Low Vision. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (Glasgow, Scotland Uk) (CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–14. doi:10.1145/3290605.3300341

[68] Yuhang Zhao, Edward Cutrell, Christian Holz, Meredith Ringel Morris, Eyal Ofek, and Andrew D. Wilson. 2019. SeeingVR: A Set of Tools to Make Virtual Reality More Accessible to People With Low Vision. <https://github.com/microsoft/SeeingVRtoolkit>

## A Example Codes

Themes	Sub-Themes	Example Codes
Difficulties of Feature Integration	A11y Issues and Needs in XR Interactions	Text illegible due to resolution; need ARIA label equivalent for XR; spatial audio for BLV; thumbstick turning harms mental map; VR motion sickness; vision-motion mismatch; no linear page flow in 3D; real-time caption challenges; lacking VR screenreader
	Sacrificing A11y for Immersion	Balance a11y and immersion; ignore guidelines for creative direction; a11y solutions could make experience worse
	Disability Representation in XR	Normative view; perception: PVI can't use VR; perception: apps don't accommodate PWD
	Performance Limitations of Standalone HMDs	Performance overhead to add a11y feature; hardware limitations; additional restrictions for mobile VR; Android vs PC environment differences
	Onboarding First-Time Users	Tag objects as interactable; video tutorial insufficient; trouble onboarding to XR; VR inexperience
Current Development Practices	Lack of Formal XR Training	Self-taught; community resources; holes in development knowledge; StackOverflow; Reddit; Discord; YouTube; Udemy; Udacity; blog posts
	Developers' Diverse Responsibilities	Developers wear many hats; survival mode; high turnover
	Dynamic Code Generation	Code easier to merge vs prefabs; dynamically create objects by script; procedurally generate data
	Reusing Code Between Projects	Reusing code; reuse components; base template scene; investing in reusable a11y frameworks
	Testing with PWD	PWD appear in general testing; difficult to find PVI testers; no access to testers; dedicated user studies
Motivations	Motivations	Time; money; audience size worth; legal; government regulations; good PR; fear of bad reviews; a11y helps people without disabilities
	Hindrances	Survival mode; MVP; a11y low priority; a11y cut to meet deadlines; conflicting priorities; limited resources; "not realistic" to add a11y
	Shared Responsibility	Accessible design; best design practices; including accessible design lowers costs; everyone has a responsibility to make accessible; no dedicated a11y developer; a11y should not be a hard sell to VR devs
Practitioners' Current XR A11y Solutions	Examples of Existing A11y Solutions	Colorblindness filters; one-handed mode; text size; UI magnification; variable height experience; visual sound indicator; volume sliders; invisible buttons
	Incorporating Alternate Input	Rewired; Xbox a11y controller; Leap Motion; Joycon; Kinect; Merge Cube; XRInput
	Hooking into System A11y Support	Custom hooks; Unity lacks a11y hooks; hook into platform functionality; screenreader; TTS; STT; invisible buttons
XR-Specific Applicability Concerns	Prior Knowledge of Guidelines	Current guidelines broad; WCAG; Microsoft's MR Standards; lack of XR guidelines; lack of awareness
	Understanding Guidelines	Not for real-time gaming; 2D not applicable to 3D; trying to fit old a11y requirements into XR; semantically ambiguous
	When to Implement a11y	A11y in design phase; a11y should start asap; easier the earlier you start; difficult to incorporate post-hoc
Preferred XR A11y Support Methods	SDK's/Toolkits	Unity XR Interaction Toolkit; MRTK; OpenXR support; Meta/Oculus toolkit
	Guideline Shortcomings	Semantically ambiguous; undefined evaluation metrics; guideline needs technical scope; WCAG is 2D-only; no guideline consensus; Microsoft's MR Accessibility Standards not fleshed out
	3rd Party Packages	Skeptical of 3rd party packages; criteria: actively developed, lightweight, open-source, easy to drop into projects, customizable
	Game Modification	Modding security concerns; modding legal problems; big tech dislikes/disallows modding; offload responsibility to community; multiplayer concerns
	Automated A11y Checking	Lacking automated a11y checks for 3D; AI a11y checking; use 3rd party a11y checker
	Guideline Improvement Suggestions	Provide examples; specific values; cost/time estimates; sample scenes; GIFs/Videos; safe vs dangerous examples; implementation demonstrations
	Need to be Directly Implementable	Easy to implement; drag and droppable plugin; plug and play; limited overhead to add a11y

Table 4: Non-Exhaustive Summary of Themes, Sub-themes, and Corresponding Example Codes

## B Beat Saber Color Options



Figure 3: We show an example from one of the most popular [26] VR games, Beat Saber [7], which offers custom color settings for a variety of virtual objects. Selecting an object (Right) like notes, lights, or wall, then picking a color from the picker, will cause the respective virtual object to change to that color. Users can create up to four custom color palettes (Left, labeled Custom 0 through Custom 3) or choose from a variety of presets themed from in-game collections (e.g., “The First”, “Origins”). As of November 2024 there do not exist built-in presets for common color vision deficiencies (e.g., protanopia, deutanopia, tritanopia) [33].