A Preliminary Interview: Understanding XR Developers' Needs towards Open-Source Accessibility Support

Tiger F. Ji*
University of Wisconsin-Madison

Yaxin Hu†
University of Wisconsin-Madison

Yu Huang‡
Vanderbilt University

Ruofei Du§
Google Labs

Yuhang Zhao¶
University of Wisconsin-Madison

Abstract

While extended reality (XR) technology is seeing increasing mainstream utilization, it is not accessible to users with disabilities and lacks support for XR developers to create accessibility features. In this study, we investigated XR developers' practices, challenges, needs when integrating accessibility in their projects. Our findings revealed developers' needs for open-source accessibility support, such as code examples of particular accessibility features alongside accessibility guidelines.

Index Terms: Human-centered computing—Accessibility—Accessibility design and evaluation methods—; Human-centered computing—Human computer interaction (HCI)—Interaction paradigms—Mixed / augmented reality; Human-centered computing—Human computer interaction (HCI)—Interaction paradigms—Virtual reality;

1 Introduction

Extended reality (XR) carries the potential to revolutionize how people communicate, interact, and collaborate within a virtual space, thus being applied to many different practical fields, such as education [13], healthcare [26], architecture [20], and accessibility [32]. However, the unique immersive and kinesthetic experience of XR has raised barriers to people with disabilities, preventing them from accessing and benefiting from these technologies. For example, users with visual impairments may lose out on essential 3D visual information [28], and users with motor impairments may not be able to complete the motion-based interactions with XR elements [17]. Furthermore, XR lacks essential accessibility support, such as screen readers for people with visual impairments or bespoke input devices for people with motor impairments, and thus excludes a large section of the global population [7].

XR designers and developers in industry play a vital role in incorporating accessibility in the mainstream development process. This means that general developers must be *motivated* and *able* to implement accessibility features in their projects. In traditional software development (e.g., web, mobile, or game development), the lack of knowledge, organizational structure, and guidance to properly implement accessibility for users with disabilities [2, 15] has been a critical issue to integrating accessibility in products. Thus, many resources and tools have been created to assist developers for accessibility integration, such as comprehensive guidelines [6] and automatic accessibility assessments for web development [22]. However, no such work exists for XR developers, who focus on an emerging technology that has no standard accessibility infrastructure and guidelines. With the increasing popularity of XR, it is important to understand developers' perspectives in accessibility, motivating and supporting them to effectively integrate accessibility into mainstream applications at an early stage.

In this paper, we aim to investigate XR developers' perspectives and needs for guidelines and tools to support accessibility integration. Specifically, we conducted a semi-structured interview study with 21 professional XR developers to understand their current attitudes, knowledge, and practices towards accessibility integration, their perceptions of the existing preliminary XR accessibility guidelines [1, 6, 21, 27], and how they could be best supported in creating accessible applications. Our participants represented a good coverage in development experience, ranging from freelancers, startups, big tech companies, and XR development platforms, thus providing a comprehensive view of XR developers' experiences and needs.

Our study revealed the current XR development practice and the unique barriers it brought to accessibility integration. With the focus of immersion, XR developers heavily relied on their first-person perspective to design and test their applications. This practice made it difficult to create accessible features as the developers lack the first-person perspective as a user with disability. Moreover, due to the immaturity of XR interactions, developers commonly followed successful examples in their development; however, current XR accessibility guidelines were too abstract and lack concrete examples and sample codes, further demotivating developers from considering accessibility. Future work will derive design implications for accessibility guidelines and tools to better support accessibility integration for XR developers.

2 Related Work

Technology for XR Accessibility. With the increasing popularity of XR, researchers have explored the experiences of people with disabilities when using the technology and identified various challenges they face [17, 29, 31]. As a result, growing efforts made in designing assistive technologies to make the applications accessible to people with disabilities. Such work mainly focuses on people with visual impairments and people with motor impairments. For people with visual impairments, both audio-based technology [12, 18] and haptic controllers [23, 30] have been designed to convey the 3D virtual environments. For people with motor impairments, works have examined what barriers are posed by operation of XR equipment and interaction with virtual objects [17, 29] or created assistive devices that allow the user to complete complex actions with a limited range of motion [5]

Despite assistive technologies being designed to facilitate XR accessibility, most of these works remain prototypes in the research field without integration into mainstream applications or platforms. Therefore, a main goal of our study is to understand how and why XR developers do or do not incorporate accessibility into their applications, and how to best support them to translate accessibility technologies from research into mainstream products.

Accessibility Guidelines for XR. Accessibility guidelines are

*e-mail: tfji@wisc.edu
†e-mail: yaxin.hu@wisc.edu
‡e-mail: yu.huang@vanderbilt.edu
§e-mail: ruofei@google.com
¶e-mail: yuhang.zhao@cs.wisc.edu

979-8-3503-4839-2/23/$31.00 ©2023 IEEE
DOI 10.1109/VRW58643.2023.00107

important resources that direct developers to implement accessibility in their development. However, XR development is a newer and more fragmented process with no widely accepted standard. While researchers have come up with general guidelines to support immersive XR experiences, they usually do not consider accessibility [4,8].

Recently, both researchers [10,16] and organizations [11,21] have started deriving accessibility guidelines that target XR experiences. For example, Heilemann et al. [10] surveyed existing accessibility guidelines for traditional video games, such as GAG, and compiled a set of guidelines that were translatable to VR games and interactions. In addition to researchers, the XR industry has paid more attention to accessibility guidelines. Oculus [21] have made a set of general accessibility guidelines to advise developers creating applications for their VR devices. Some accessibility interest groups have also created considerations for accessible XR design [11], but typically do not provide actionable guidelines or specific feature examples.

Current XR accessibility guidelines are in their infancy, with no rigorous validation or broad agreement. They lack technical details to guide developers to implement these guidelines. Therefore, it was important to obtain the developers' perspective of existing XR accessibility guidelines to understand their interpretations and needs when applying the guidelines in their projects.

Development Tools for XR Accessibility. To support XR development, a variety of tools have been created, usually aimed at simplifying or automating common functionalities. For example, the XR Interaction [25] Toolkit provided prefabs that allowed for cross platform compatibility with various VR devices and scripts for basic VR interactions. Nebeling [19] also reviewed recent XR development tools, noting trends in provided functionalities and interfaces such as increasing affordance for developers familiar with non-XR tools.

Little research has designed tools to support accessibility integration in XR. The only work, to our knowledge, is Zhao et al. [31] that designed a developer toolkit with 14 low vision tools to help developer incorporate the low vision tools in their VR applications. Six Unity developers were interviewed to evaluate the toolkit. However, this work mainly focused on the creation and evaluation of the toolkit. More in-depth investigations are needed to explore XR developers' experiences, barriers, and needs in accessibility integration, thus designing more suitable support to motivate and facilitate this process.

3 Methodology

The goal of our research is to identify what barriers prevent XR developers from implementing accessibility features and how they can best be supported to do so. We recruited 21 XR developers (17 male, 4 female), aged 18-58. Our participants came from diverse organizations, ranging from freelancers to small startups, to large software companies (e.g., Microsoft, Apple, Google, Amazon), to XR platforms (e.g., Unity, Unreal, CryEngine). All participants had relatively rich XR development experience (1.5-10 years, $mean = 4.1, SD = 1.88$). Participants had developed on different XR platforms (e.g., Unity, Unreal, A-Frame, Babylon React Native) and for different XR devices (e.g., Oculus Quest, HTC Vive, HoloLens).

We conducted a semi-structured interview that lasted approximately for two hours. The study was conducted remotely through Zoom video calls. Participants were asked to prepare a project that they could screen share to demonstrate their development practices. We asked participants about their XR development experience, such as what projects or features they had worked on, and if they had any experience developing accessibility features. They then detailed development cycles and personal coding practices for XR, screen-sharing when possible to provide visual clarification for described practices. Moreover, participants were asked about the usage of any third-party support in their XR development, such as libraries, APIs, plugins, or assets, and why they used those tools.

We further explored participants' interpretation of state-of-the-art XR accessibility guidelines. We collected a set of accessibility guidelines by searching online with the keywords "Augmented Reality (or AR)," "Virtual Reality (or VR)," "Mixed Reality (or MR)" "Extended Reality (or XR)," and "game" with "accessibility guidelines," and took the results found on the first five pages of Google results, as well as any guidelines from prior literature [10]. Four sets with an equal number of guidelines were created based on the type of disabilities, including motor, visual, cognitive, and speech/hearing disabilities. We selected the most frequently occurring guidelines for each group and presented each participant with one set of guidelines, discussing their interpretation of the guidelines and potential challenges and needs for implementation. We ended the study by asking participants about how much time and effort was viable to dedicate to accessibility, their greatest barriers to accessibility implementation, and what assistance (e.g., tools, plugins, guidelines) could best support their accessibility integration process.

We analyzed the transcripts through thematic analysis [3], starting with open coding between two researchers. Following axial coding, affinity diagramming was used to identify connections and relations between codes.

4 Findings

We present the key findings on the unique XR development practices and the barriers they brought to accessibility integration.

Lacking the First-Person Perspective of Users with Disabilities. Due to the complexity of the immersive user experience in XR projects, all participants relied on their own first-person perspective or personal experience to design and test their XR interactions. Even for larger scale companies that had testing frameworks for XR projects, such as automated code testing or peer review, developers still personally tested their work to determine whether the intended immersive experience was achieved. When discussing her thought process while designing and testing for her sword fighting game, P2 (gender: female, age range: 31-40, organization type: freelance) explained how important her personal experience with sword fighting was to development and how difficult it was to imagine what problems other people might have: *"I think that because [XR]'s so unintuitive and so complex, you really have to see people playing it more so than like, a simple mobile game...It's really hard to guess what's gonna happen wrong when you put that headset on a different person...This is what I've spent all my time in, working on this and working on other virtual reality applications, so I'm very inured to it...But without seeing [the experience] from the [user] viewpoint, it's hard to tell how exactly the experience is going to go."*

Seven participants noted difficulty when imagining accessibility features for users with disabilities, as they did not have disabilities or an adequate understanding of how users with disabilities would experience their applications. For example, when discussing audio descriptions for virtual objects, P6 (male, 31-40, freelance) expressed uncertainty about what functionality would be needed and how that would interact with the rest of his project. He explained: *"Well, I imagine somebody is completely blind...But imagining how this interface actually gets used for a completely visually impaired person is really hard for me to picture. How are you going to aim your laser at something and select it [to hear the audio description]?"*

Lack of Trust in Third-Party Tools. Without a reputable organization supporting an accessibility tool, six participants would not trust the tool's reliability enough to adopt it as their standard practices. Three participants noted that they would first need to spend time to thoroughly understand how every part of the tool works to avoid compatibility or security issues . This was especially true for XR development due to the frequent updates in the development platform. Three participants pointed out that many useful XR development tools have been used before but dropped due to the lack of maintenance by the third-party creators. Developers then had to update those tools themselves or recreate the functionality from scratch. For example, P7 (male, 18-30, freelance) noted the trouble he encountered when trying to find third-party plugins to add compatibility with more VR devices: *"One of the things we've noticed is, we have tried to use [third-party] plugins or whatever in order to expand the number of VR controllers and VR headsets. There are good options out there, but then no one's maintaining them and become outdated, or they're still missing some key features. So something in-house is generally what we found to be better in terms of the design process."*

Using third party tools could be especially risky for larger tech companies with large-scale projects and a large number of users. For instance, P12's (male, 31-40, startup) organization, which was beginning to scale up and need accessibility frameworks for their projects, also preferred to keep all codes controlled internally. He explained: *"We don't use as much third party stuff, because it'll work, like you use it, but then it updates, and then all of a sudden breaks everything. And you have people with all these contracts, and all of a sudden their things aren't working anymore.* As such, five participants specifically suggested XR development platforms to create the tools and frameworks for accessibility, to assure developers of continued support and compatibility.

Valuing Platform Generalizability. Unlike traditional 2D applications with more uniform platforms and interactions, such as PCs or smartphones, XR users also adopted more diverse XR headsets with different controllers and inputs. As a result, XR developers placed high value on the generalizability of their applications . Seven participants emphasized the importance of cross platform compatibility for reaching as wide of an audience as possible. One of P3's (male, 18-30, large company) web XR projects was even rewritten three times, ending up with utilizing cross-platform programming languages and libraries—React and Babylon Native—for their inherent compatibility with multiple platforms. As P3 explained, *"For the mobile mixed reality, we actually wrote it three times... And then we rewrote it in our own thing called Babylon Native... it's basically a wrapper type script, the native renders that Android and iOS use. And then from there, it also supports mixed reality so you get the camera plus you get the 3D render of Babylon Native, and voila, you suddenly have cross platform mobile mixed reality."*

The seven participants perceived the lack of a consistent accessibility framework across all platforms as a barrier to making accessibility features that were compatible with those platforms. Extra effort would be needed to implement features on each supported platform. For example, when discussing custom fonts for users with visual impairments, P9 (male, 18-30, freelance) noted that the potential incompatibility with different platforms should be addressed when designing accessible fonts. He explained: *"You need to make [the custom font] programmatically supported by different [XR] operating systems and software libraries, because these libraries are simply not up to standard in a lot of cases...You're making something work for all operating systems, I think back end development needs to be done very early in order to make [custom fonts] feasible. Then front end developers and designers can probably work together to determine how to create adaptable interface elements."*

Desires for Implementation Examples of XR Accessibility Features. Eleven participants expressed the need for concrete implementation examples of accessibility features in XR and how users with disabilities might interact with the features. We found that many XR developers adopted an example-driven approach when designing and implementing XR projects. They usually referred to existing successful examples for ideas during the design phase. With the examples, developers could try the interactions in person and gained the first-hand experience to reuse or re-purpose the example for their own XR project. Three participants referenced specific applications when determining the implementation of their XR features. For example, P16 (male, 18-30, platform) regularly surveyed existing applications before designing features. He reported: *"For our technical design, I basically study some state-of-the-art VR games and also study other company's solutions, like Meta or Microsoft's HoloLens toolkit, and we kind of figure out some best approaches."*

However, participants were unaware of any applications or example projects for XR that they could reference for accessibility features for users with disabilities. Therefore, when viewing the description-based XR accessibility guidelines, most participants had no concrete vision for how an accessibility feature should be implemented. They would need to experiment or conduct more iterations of their prototypes to identify a working implementation. P15 (male, 18-30, large company) emphasized the importance of having implementation examples in conjunction with accessibility guidelines: *"I think [an implementation example] is super important. If I'm building a VR app with some kind of locomotion, I would just like to see how to do it. Maybe not even separate for different impairments, but at least the proper way that will cover the most [disabilities]...The more such components there is, the better. Not everything can be covered, that's why the guidelines have to exist, but I think this will build some general understanding very well."*

Standardization of Assistive XR Hardware. Besides software solutions, participants also discussed the integration of custom XR devices for accessibility purposes. Following the idea of programming for general compatibility, many participants used plugins, such as the XR Interaction (XRI) Toolkit [25] or the Virtual Reality Toolkit (VRTK), that mapped common XR interactions like grabbing objects or locomotion to XR input standards (e.g., OpenXR, Windows Mixed Reality) [9, 14]. Developers would be saved the trouble of learning and coding for the APIs of each standard while obtaining compatibility with any device using those input standards. When discussing various hardware-based assistive technologies [23, 30] and their potential integration to the mainstream XR setup, six participants suggested a standardized input mapping framework for custom devices or a plugin that automatically enabled the compatibility for these devices. They believed that developers would not keep up with state-of-the-art assistive XR devices if each device required a different implementation. P13 (male, 31-40, freelance) stated: *"If [users with disabilities] can't use these [standard] controllers, then you're looking at manufacturing custom hardware. And it's sort of like, who's going to do that? So that device almost has to be there already. And then you need to have scripting pre-made to the extent that you can just integrate it with, like, the XRI toolkit and Unity...I think it would be a huge barrier to have to ask the studio to design for custom hardware devices or something like that."* Some participants also pointed to mouse/keyboard and joystick controllers as examples of devices commonly supported in games due to their standardized and thus easy to implement inputs.

5 Conclusion & Future Work

Our research contributed the first exploration of XR developers' practices, barriers, and needs when implementing accessibility for users with disabilities in mainstream applications or platforms. Our study highlighted the unique XR development practices and the corresponding challenges in accessibility integration, including (1) The first-person testing practice for immersive interactions made it difficult to design accessible features since the developers needed but lacked the first-person experience from people with disabilities; (2) The lack of maintenance of third-party XR development tools due to the frequent platform updates made it hard for developers to trust and adopt third-party assistance; (3) The device diversity increased the importance as well as the difficulty of designing cross-platform accessibility features; (4) developers followed an example-driven practice but current XR accessibility guidelines did not provide concrete code examples and technical guidance; (5) State-of-the-art custom assistive devices from research lack easy integration into mainstream XR setups.

Many of these challenges are caused by the general developer's lack of awareness of state-of-the-art accessibility techniques and the perspectives of people with disabilities. This presents the opportunity for open source solutions that consolidate the collective knowledge of the accessibility community. For example, Thiel & Steed [24] proposed a metric to evaluate the physical accessibility of VR games based on data collected from actual players. Similar metrics for other disabilities, such as visual or hearing, could be made with data sourced from the accessibility community or research, supporting developers to validate accessibility designs.

While our current study has drawn preliminary insights in XR developers' practices and challenges in development and accessibility integration, more thorough analysis will be conducted to compare the attitudes and needs of developers from different types of organizations. Moreover, based on our findings, we will derive design implementation to inspire more actionable XR accessibility guidelines and more useful development tools, motivating and enabling XR developers to easily design and implement accessibility features in mainstream applications.

References

[1] AbleGamers. Accessible Player Experiences (APX), 2022. doi: 10.1007/978-3-319-94277-40

[2] T. Bi, X. Xia, D. Lo, J. Grundy, T. Zimmermann, and D. Ford. Accessibility in Software Practice: A Practitioner's Perspective. *ACM Trans. Softw. Eng. Methodol*, 31(4), jul 2022. doi: 10.1145/3503508

[3] V. Braun and V. Clarke. *Thematic Analysis*. American Psychological Association, 2012.

[4] G. Catak, S. MASALCI, and S. Şenyer. A Guideline Study for Designing Virtual Reality Games. *AJIT-E: Online Academic Journal of Information Technology*, 11:12–36, 12 2020. doi: 10.5824/ajite.2020.04.001.x

[5] I. Choi, E. Ofek, H. Benko, M. Sinclair, and C. Holz. 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*, pp. 1–13, 2018. doi: 10.1145/3173574.3174228

[6] B. Ellis, G. Ford-Williams, L. Graham, D. Grammenos, I. Hamilton, E. Lee, J. Manion, and T. Westin. A Straightforward Reference for Inclusive Game Design, 2012.

[7] C. for Disease Control and Prevention. Disability Impacts All of Us, 2022. last accessed 15 Jun 2022.

[8] S. Garcia Fracaro, P. Chan, T. Gallagher, Y. Tehreem, R. Toyoda, K. Bernaerts, J. Glassey, T. Pfeiffer, B. Slof, S. Wachsmuth, and M. Wilk. Towards Design Guidelines for Virtual Reality Training for the Chemical Industry. *Education for Chemical Engineers*, 36:12–23, 2021. doi: 10.1016/j.ece.2021.01.014

[9] K. Group. OpenXR, 2022. last accessed 15 Jun 2022.

[10] F. Heilemann, G. Zimmermann, and P. Münster. Accessibility Guidelines for VR Games - a Comparison and Synthesis of a Comprehensive Set. *Frontiers in Virtual Reality*, 2, 2021. doi: 10.3389/frvir.2021.697504

[11] X. Ignite. Designing XR for Accessibility and Inclusion, 2022.

[12] T. Ji, B. R. Cochran, and Y. Zhao. VRBubble: Enhancing Peripheral Awareness of Avatars for People With Visual Impairments in Social Virtual Reality. In *The 24th International ACM SIGACCESS Conference on Computers and Accessibility*, number 3 in ASSETS '22. Association for Computing Machinery, 2022. doi: 10.1145/3517428.3544821

[13] S. Kavanagh, A. Luxton-Reilly, B. Wuensche, and B. Plimmer. A Systematic Review of Virtual Reality in Education. *Themes in Science and Technology Education*, 10(2):85–119, 2017. doi: 10.1109/ICWT47785.2019.8978263

[14] Microsoft. Microsoft Mixed Reality, 2022. last accessed 15 Jun 2022.

[15] D. Miranda and J. a. Araujo. Studying Industry Practices of Accessibility Requirements in Agile Development. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, SAC '22, p. 1309–1317. Association for Computing Machinery, New York, NY, USA, 2022. doi: 10.1145/3477314.3507041

[16] M. Mott, E. Cutrell, M. Gonzalez Franco, C. Holz, E. Ofek, R. Stoakley, and M. Ringel Morris. Accessible by Design: An Opportunity for Virtual Reality. In *2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pp. 451–454, 2019. doi: 10.1109/ISMAR-Adjunct.2019.00122

[17] M. Mott, J. Tang, S. Kane, E. Cutrell, and M. Ringel Morris. "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*, ASSETS '20. Association for Computing Machinery, New York, NY, USA, 2020. doi: 10.1145/3373625.3416998

[18] V. Nair, J. L. Karp, S. Silverman, M. Kalra, H. Lehv, F. Jamil, and B. A. Smith. NavStick: Making Video Games Blind-Accessible Via the Ability to Look Around. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pp. 538–551, 2021.

[19] M. Nebeling. XR Tools and Where They Are Taking Us: Characterizing the Evolving Research on Augmented, Virtual, and Mixed Reality Prototyping and Development Tools. *XRDS*, 29(1):32–38, oct 2022. doi: 10.1145/3558192

[20] M. Noghabaei, A. Heydarian, V. Balali, and K. Han. Trend Analysis on Adoption of Virtual and Augmented Reality in the Architecture, Engineering, and Construction Industry. *Data*, 5(1):26, 2020.

[21] Oculus. Designing Accessible VR, 2022.

[22] Pa11y. Pa11y, 2022. last accessed 15 Jun 2022.

[23] A. F. Siu, M. Sinclair, R. Kovacs, E. Ofek, C. Holz, and E. Cutrell. Virtual Reality Without Vision: A Haptic and Auditory White Cane to Navigate Complex Virtual Worlds. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, p. 1–13. Association for Computing Machinery, New York, NY, USA, 2020. doi: 10.1145/3313831.3376353

[24] F. J. Thiel and A. Steed. Developing an accessibility metric for vr games based on motion data captured under game conditions. *Frontiers in Virtual Reality*, 3, 2022. doi: 10.3389/frvir.2022.909357

[25] Unity. XR Interaction Toolkit, 2022. last accessed 15 Jun 2022.

[26] R. M. Viglialoro, S. Condino, G. Turini, M. Carbone, V. Ferrari, and M. Gesi. Augmented Reality, Mixed Reality, and Hybrid Approach in Healthcare Simulation: A Systematic Review. *Applied Sciences*, 11(5):2338, 2021.

[27] W3C. XR Accessibility User Requirements, 2020. doi: 10.1109/CAMAD.2018.8514957

[28] D. A. Waters and H. H. Abulula. The Virtual Bat: Echolocation in Virtual Reality. In *Proceedings of the 2001 International Conference on Auditory Display*. Espoo, Finland, 2001.

[29] M. Yamagami, S. Junuzovic, M. Gonzalez-Franco, E. Ofek, E. Cutrell, J. Porter, A. Wilson, and M. Mott. 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):1–25, 2021. doi: 10.1145/3510463

[30] Y. Zhao, C. L. Bennett, H. Benko, E. Cutrell, C. Holz, M. R. Morris, and M. Sinclair. Enabling People With Visual Impairments to Navigate Virtual Reality With a Haptic and Auditory Cane Simulation. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, p. 1–14. Association for Computing Machinery, New York, NY, USA, 2018. doi: 10.1145/3173574.3173690

[31] Y. Zhao, E. Cutrell, C. Holz, M. R. Morris, E. Ofek, and A. D. Wilson. 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*, CHI '19, p. 1–14. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3290605.3300341

[32] Y. Zhao, S. Szpiro, J. Knighten, and S. Azenkot. CueSee: Exploring Visual Cues for People With Low Vision to Facilitate a Visual Search Task. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '16, p. 73–84. Association for Computing Machinery, New York, NY, USA, 2016. doi: 10.1145/2971648.2971730