SceneGenA11y: How can Runtime Generative tools improve the Accessibility of a Virtual 3D Scene?

Xinyun Cao
Computer Science and Engineering
University of Michigan
Ann Arbor, Michigan, USA
xinyunc@umich.edu

Kexin Phyllis Ju
School of Information
University of Michigan
Ann Arbor, Michigan, USA
kexinju@umich.edu

Chenglin Li
University of Michigan
Ann Arbor, Michigan, USA
lchengl@umich.edu

Dhruv Jain
Computer Science and Engineering
University of Michigan
Ann Arbor, Michigan, USA
profdj@umich.edu

Abstract

With the popularity of virtual 3D applications, from video games to educational content and virtual reality scenarios, the accessibility of 3D scene information is vital to ensure inclusive and equitable experiences for all. Previous work include information substitutions like audio description and captions, as well as personalized modifications, but they could only provide predefined accommodations. In this work, we propose SceneGenA11y, a system that responds to the user's natural language prompts to improve accessibility of a 3D virtual scene in runtime. The system primes LLM agents with accessibility-related knowledge, allowing users to explore the scene and perform verifiable modifications to improve accessibility. We conducted a preliminary evaluation of our system with three blind and low-vision people and three deaf and hard-of-hearing people. The results show that our system is intuitive to use and can successfully improve accessibility. We discussed usage patterns of the system, potential improvements, and integration into apps. We ended with highlighting plans for future work.

CCS Concepts

• **Human-centered computing** → Accessibility; Accessibility systems and tools.

Keywords

accessibility, generative AI, BLV, DHH, virtual 3D scenes

ACM Reference Format:
Xinyun Cao, Kexin Phyllis Ju, Chenglin Li, and Dhruv Jain. 2025. SceneGenA11y: How can Runtime Generative tools improve the Accessibility of a Virtual 3D Scene?. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA '25), April 26–May 01, 2025, Yokohama, Japan.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3706599.3720265

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
CHI EA '25, Yokohama, Japan
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1395-8/2025/04
https://doi.org/10.1145/3706599.3720265

1 Introduction

Virtual 3D environments are commonly used for games, social apps, 3D cinematics, designer tools, and educational apps for features like simulating real-world experiences (e.g. [34]) and visualizing 3D information (e.g. [7]). Mixed reality applications further enable six degrees of freedom movement [27] and new interaction methods [5]. However, these innovations introduce accessibility challenges. For example, existing screen readers may be incompatible with the user interface of a virtual reality game [26], and traditional captions might fail to convey the locations of spatial sound sources in a 360 degree movie [64].

Prior work includes different methods to improve the accessibility of virtual 3D scenes. For commercial 3D video games, industry standards [20, 28] provide guidelines for pre-defined accessibility settings in commercial apps [6, 61]. More novel toolkits explore substituting inaccessible modalities in specific contexts, like substituting visuals with sound [43, 65, 66] and haptics [24, 49, 58], or substituting auditory information with visuals [23, 32] and haptics [23, 67]. In addition to substitution, newer toolkits enable users with partial vision or hearing to modify the visual [59] and auditory [11] information presentation based on their ability. Despite the diversity of prior work, they cannot support on-the-go accessibility improvements, meaning all the information substitutions and modifications must be predefined by the developers. As researchers and users with disabilities advocate for more personalized accessibility technology [8, 17, 46], this calls for a more dynamic solution.

Building on top of past work, this paper proposes SceneGenA11y, a tool that responds to user's natural language prompts to improve accessibility of a 3D virtual scene in runtime. The system primes the LLM agent with accessibility knowledge and best practices to interpret related prompts. The users can explore the items in the scene through point-and-click or LLM query. They can also use the tool to iteratively create modifications in the scene and verify them. We compiled eight accessibility categories the system can improve on, available to users as guidelines. Four categories address the visual dimension, including changing color, object location, size, and brightness. The other four address the auditory dimension, including changing volume, pitch, spatial range, as well as transcript understanding.

To understand the usability of the prototype, we conducted a preliminary evaluation of the system with three blind and low vision (BLV) users and three deaf and hard-of-hearing (DHH) users. We employed two scenarios: an indoor scene with numerous visual elements and a social scene with extensive conversations. The users were encouraged to freely utilize our system within the scenes to make the experience more accessible to them. We collected ratings of the system and conducted semi-structured interviews. We report on the findings of our evaluation, including system usability, usage patterns, and insights for app integration, and end the paper with our plans for future work.

The contributions of this work include: 1) SceneGenA11y prototype, an LLM system for virtual 3D environments that responds to user's accessibility-related questions and modification requests in runtime; 2) a preliminary evaluation of the usability of the system.

2 Related Work

Our design of the system is grounded in past work about accessibility in 3D environments and Generative AI (GenAI) tools to improve accessibility. Our system is built on prior work in runtime generative AI tools for virtual 3D scenes.

2.1 Accessibility in Virtual 3D Environments

Here we review past work in 3D environment accessibility for BLV and DHH people.

For BLV users, the majority of previous work focuses on audio description [35, 43, 65, 66, 68–70] and haptic feedback [24, 31, 44, 49, 58]. The Scene Weaver prototype by Balasubramanian *et. al.* proposed giving BLV users the agency to choose when and how to perceive the environment [68]. The Canetroller model introduced by Zhao *et. al.* successfully allowed BLV users to navigate and understand a Virtual 3D scene using a haptic white cane [58]. SeeingVR enhanced the accessibility of virtual 3D scenes for low-vision users with 14 tools, including magnification, contrast increase, and recoloring [59]. Commercial 3D games provide different kinds of accessibility settings for BLV players, including text-to-speech, high contrast display, lock aim, and navigation assistance [6, 71].

For DHH users, past work mainly includes substituting sound events with visual [23, 32, 36, 37] and haptic feedback [23, 36, 67] in a 3D scene. The ImAc project investigated the use of subtitles, audio descriptions, and sign language in immersive TV production [37]. Jain *et. al.* [23] explored visual and haptic alternatives of sounds in VR. Mirzaei et al. evaluated a multi-modal 3D immersive system consisting of audio, visual elements indicating sound sources, and haptic feedback like in-ear motors [36, 67]. SoundModVR by Cao *et. al.* modified sounds to enhance 3D accessibility for the hard of hearing [11]. In commercial 3D games, developers have included features like captions [52], directional hints of sound source [60], and volume adjustment for different sound groups [6] to improve the sound accessibility of their games.

Previous work in these areas guided the design of our system, including the object identification feature and the categories of changes and queries users can use to understand and modify the scene. On the other hand, prior work is limited to accessibility settings predefined by the developer. As researchers stress the importance of context-aware and personalized accessibility [8, 17, 46], this work focuses on on-the-go runtime accessibility improvements prompted by the user.

2.2 Generative AI for Accessibility

Visual Language Models (VLM) models [2, 72] and Audio Language models [30] can be deployed to support sensory disabilities. Past work to support BLV users have delivered visual descriptions for images [56], real-life scenario [13, 73] and data visualization [19, 45]. While past work for DHH people has used LLM to improve accuracy of ASR [16] and CART [53], as well as context-aware sound interpretation [30].

Past work has also used Generative AI's code generation ability and knowledge of Web Content Accessibility Guidelines [74] to improve web accessibility. Mowar *et. al.* studied developers untrained in web accessibility creating web UIs with and without AI-powered Copilot, finding its impact on web accessibility improvement limited by the need for expertise input [38]. Aljedaani *et. al.* [4] and Othman *et. al.* [40] demonstrated that LLMs can remediate non-accessible web code and generate inclusive code with proper supervision and rectification. Kodandaram and Uckun *et. al.*'s Savant system [29], a universal web interface powered by LLMs, allowed BLV users to navigate the web using natural language. These works inspired our approach of using LLM-generated code to enhance the accessibility of 3D environments.

Despite the growing popularity of Generative AI in accessibility tech, its use in accessibility raises several concerns. Firstly, GenAI lacks comprehensive accessibility knowledge. In an autoethnographic paper by Glazko *et. al.* [18], users reported that LLMs could "parrot back" accessibility rules but struggled to apply them. Another concern is the biases and ableist stereotypes introduced through the training materials of the LLM [18, 50]. Additionally, because of the randomness of LLMs and the potential of hallucination [57], an AI accessibility system needs to provide a robust and convincing verification system and methods for users to contest if they disagree [1, 3].

Our system is grounded in the understanding of past work in Generative AI for accessibility. The scope of this work focuses on sensory disability, including BLV and DHH users. We use LLM-generated code to address accessibility issues, incorporating prompt engineering and a verification loop to overcome LLM limitations, such as gaps in accessibility understanding and hallucinations.

2.3 Generative AI for Virtual 3D Scenes

Prior work has explored GenAI's ability for scene creation and editing [21, 55, 75]. GenAI systems were also built to generate code and compile novel behavior based on natural language input. Jennings *et. al.* introduced GROMIT, an LLM-based runtime behavior generation system [25]. The system feeds a Unity scene in WSON format to an LLM, allowing users to prompt the LLM to generate code for a behavior that is then compiled at runtime. Their work explored this system in game development and showcased its potential to create new game mechanics through a user study with developers. Another example is the LLMR system proposed by De La Torre *et. al.*, which is a framework for the real-time creation and modification of

interactive Mixed Reality experiences using LLMs [15]. This system employs techniques like scene understanding and task planning to produce and edit diverse objects, tools, and scenes. De La Torre *et. al.* discussed the potential of this method to improve accessibility, such as adapting for color blindness, near-sightedness, or design for children. However, the LLMR paper lacks a comprehensive exploration of accessibility modifications achievable by the system, and its usability for accessibility has not been fully evaluated through a user study.

Prior work has shown the capability of LLM to understand 3D scenes semantically and generate models, behavior, and modifications. Built on top of the GROMIT [25] system, our system is specifically designed for using GenAI to navigate, query, and modify 3D scenes for accessibility. Our work also includes an evaluation of this system for accessibility.

3 SceneGenA11y System Design

The SceneGenA11y system used the open-sourced GROMIT toolkit [25, 39] to achieve runtime generation and compilation in Unity 3D. It receives the WSON representing the semantic scene graph of the scene and a natural language prompt and compiles the resulting code. To enable auditory understanding, developers can incorporate sound information as text descriptions into the WSON. The LLM model used in the evaluation is GPT-4o.

3.1 Categories of Accessibility Improvements

The system's prompt interface functions as a natural language tool, enabling users to articulate their needs freely. However, to provide structure, the system is designed to support a list of categories informed by prior research in 3D accessibility technology for DHH and BLV individuals. Four categories focus on the visual dimension, while four address the audio dimension. Individuals from either group may benefit from prompts related to both dimensions.

Change Color: The tool can be utilized to modify the color or color scheme of item(s). Color adjustments can aid individuals with low vision in object recognition [54], while modifying color palettes can enhance accessibility for colorblind users [41].

Change Object Location: The tool can facilitate the movement of objects, characters, and players, streamlining object retrieval and navigation. This category was inspired by existing voice-activated game mod tools [76].

Change Sizes: The tool allows for adjusting the size of objects and text fonts. Previous studies demonstrated that enlargement can enhance visual prominence and ease of identification for low-vision users [42, 51].

Change Scene Brightness: For low-vision users, low brightness may hinder visibility, while excessive brightness can be overwhelming [59]. Our tool can make the scene or certain light sources brighter or dimmer.

Change Volume: The tool enables users to adjust the volume of sound source(s). This category builds on prior research demonstrating the utility of focusing on specific sounds for both BLV [12, 62] and DHH individuals [11, 47].

Change Pitch: The tool allows users to shift the frequency of certain sound source(s), which could make sounds more distinguishable for BLV [12] and hard-of-hearing people with frequency-specific hearing loss [22, 33].

Change Spatial Range: Spatial audio is an important component in 3D spatial perception [63]. The tool enables users to adjust spatial range of sounds, enhancing spatial awareness and selective attention.

Transcript Understanding: LLMs demonstrate strong text processing capabilities [48], which the tool utilizes to summarize character dialogue transcripts and enable users to query and modify based on these summaries.

3.2 System Workflow

The system uses prompt engineering to improve its ability to address accessibility-related prompts. For users, the system workflow includes the following steps:

1. The users interact with a multi-modal object identification interface to understand the objects in a scene.
2. The users ask questions in a specific category about the identified object(s).
3. The users prompt the tool to perform changes in a category to improve accessibility.

To assist with steps 2 and 3, the users are provided with documentation of supported modifications and queries. The users can return to previous steps in an iterative process.

3.3 Accessibility-Specific Prompt Engineering

Our system primes the LLM agent to perform modifications and respond to queries related to the categories outlined in Section 3.1. It is achieved through static prompt engineering and dynamic prompt engineering. The static priming message includes a set of accessibility-related rules specific to the development environment (Unity3D), like how to change color and color palette or use spatial relationships. Dynamic prompt engineering involves updating information needed to represent changes in a scene of an object in the WSON data, like color (in HEX code) or volume of audio source. When the user enters an LLM prompt, the system constructs a message to the LLM including dynamically primed WSON data, the prompt, previous prompt history, and the static priming message. See Appendix A1 and A2 for the priming message and code used.

3.4 Multi-modal Object Identification

The object identification process allows users to understand the viable objects in a scene. It includes features: item selection and important item retrieval. Figure 2(a) shows their corresponding UI. The multimodal item selection includes two methods: point-and-click and natural language. For the former, users can use the mouse to point and click on an object to select it. For the latter, users can use the LLM prompt system and the format "select . . ." to select an object based on natural language description. Once an object is selected, the system then displays its information: its name, description, and its location relative to the player, which is calculated by a function, not the LLM. The system also supports bookmarking an object and retrieving its information later. If the scene has changed, the information retrieved would be dynamic.

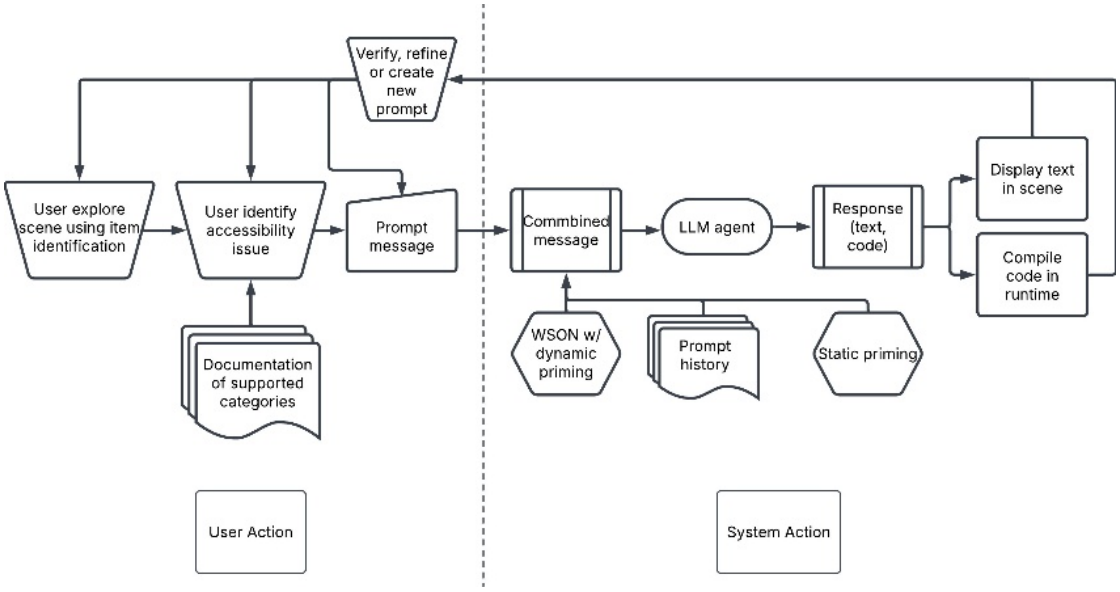
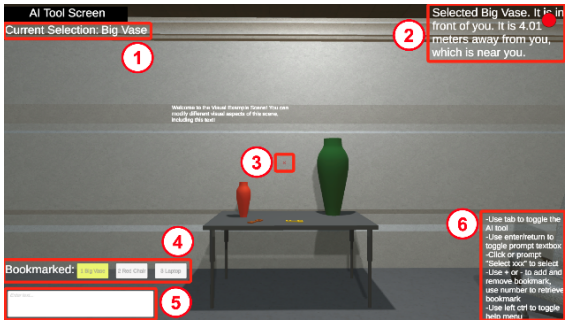
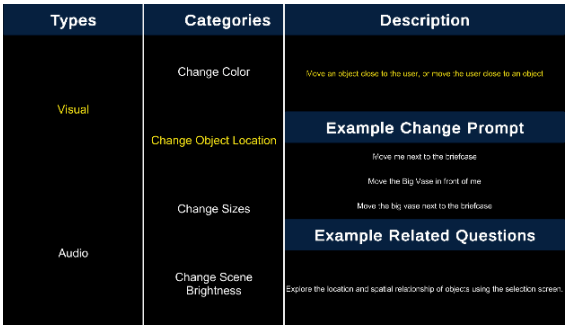


Figure 1: System workflow diagram

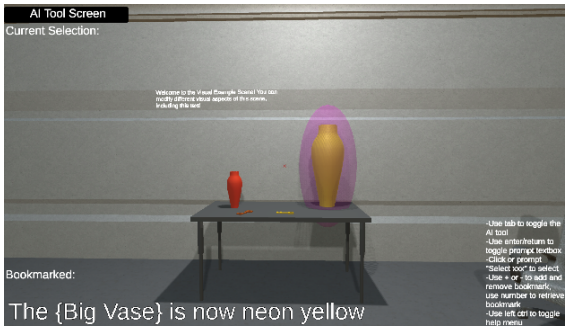


(a)

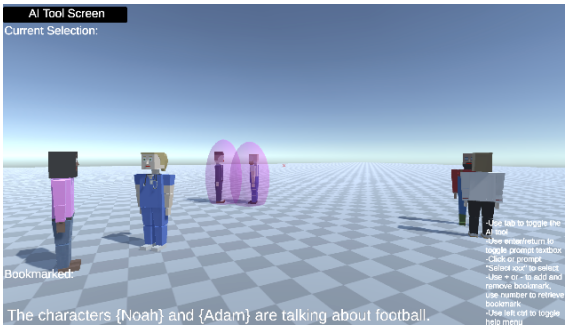


(b)

Figure 2: (a) The UI of the AI tool. The red border and numbers are graphic overlay. (1) shows the current selection; (2) shows the description of the item currently selected; (3) is the anchor for point-and-click; (4) is the bookmark list; (5) is the prompt text entry box; and (6) is the cheat sheet for controls of the tool. (b) The UI of the help menu.



(a)



(b)

Figure 3: Example results of modification and query prompts. (a) result of the prompt "Make the big vase neon yellow". (b) result of the prompt "Who are talking about football?".

Table 1: DHH participant demographics information. The hearing loss levels were self-identified. "LHLL" stands for "Left Ear Hearing Loss Level"; "RHLL" stands for "Right Ear Hearing Loss Level"; "PMoC w/DHH" stands for "Preferred Mode of Communication with DHH people"; "PMoC w/Hearing" denotes "Preferred Mode of Communication with Hearing people; "Exp w/ Video Games" means "Experience with playing Video Games"; "Exp w/ LLMs" means "Experience with using tools powered by Large Language Models".

ID	Age	Gender	Identity	LHLL	RHLL	Onset Age	PMoC w/DHH	PMoC w/Hearing	Exp w/ Video Games	Exp w/ LLMs
P1	25	Men	Hard of hearing	Profound: 91 dB or more	No loss	8 yrs	Writing	Verbal	Frequently	Expert
P2	67	Men	Hard of hearing	Severe: 71-90 dB	Severe: 71-90 dB	7 yrs	Verbal	Verbal	A few	Never
P6	24	Men	Deaf	Profound: 91 dB or more	Profound: 91 dB or more	5 yrs	Writing	Writing	Frequently	Frequently

Table 2: BLV participant demographics information. The vision loss levels were self-identified. "LVLL" stands for "Left Eye Vision Loss Level"; "RHLL" stands for "Right Eye Vision Loss Level"; "Exp w/ Video Games" means "Experience with playing Video Games"; "Exp w/ LLMs" means "Experience with using tools powered by Large Language Models".

ID	Age	Gender	LVLL	RVLL	Onset Age	Exp w/ Video Games	Exp w/ LLMs
P3	43	Men	Low Vision	Low Vision	Birth	A few	A few
P4	35	Women	Blind	Blind	Birth	Regularly	Regularly
P5	44	Men	Light perception	Light perception	30 yrs	A few	Never

3.5 LLM-supported Query and Modification Loop

The tool supports queries and modifications along the dimensions and categories listed in Section 3.1. The user inputs both types of prompts into a textbox and submits them to the system. For query prompts, the system responds with text-based answers. For modification prompts, the system compiles and executes the code and then displays an explanation of the actions performed. The prompt interface is shown in Figure 2(a), and the example results are shown in Figure 3. The system facilitates a query-modification loop, allowing users to request modifications within a category, verify the changes through queries, and iteratively refine with additional queries and modifications. This verification loop has proven important for user confidence [1, 18]. The system features an interactive help menu that documents the two dimensions and eight categories, providing explanations, example modification prompts, and example query prompts for each category. See Figure 2(b) for the UI of the help menu.

4 Preliminary Evaluation

To assess our system design and workflow, we conducted a preliminary scene-based evaluation of the ScenGenA11y system. This study aimed to explore **RQ1**: Whether our system could help people with disability to better experience a 3D scene, **RQ2**: What kind of prompts would be helpful for them, and **RQ3**: What are the main concerns surrounding using LLM tools to produce accessibility improvements.

4.1 Participants

We recruited three DHH participants and three BLV participants through mailing lists and word of mouth. DHH participants (two hard of hearing and one Deaf) were 24 to 67 years old (*mean*=38.7, *SD*=20.0). Two had profound to severe hearing loss and one had unilateral hearing loss. All three participants had prior experience with video games, and some participants used their accessibility features. One participant identified himself as an expert in using LLM and one participant had never used LLM tools before. BLV participants (two low vision and one blind) were 35 to 44 years old (*mean*=0.7, *SD*=4.03). Two participants had experience with using audio cues in video games to enhance accessibility. Only one participant had used LLM, while one participant had used LLM only a few times, and one participant had never used LLM before.

4.2 Scenes

We designed two different scenes: (i) Room scene and (ii) Social scene for participants to explore. SceneGenA11y is implemented in each scene. The **Room scene** is designed to evaluate our system in an environment that primarily focuses on the visual aspect. The scene is set inside a spaceship. The room contains various objects arranged with spatial complexity, some of which emit audio. The

Social scene aims to evaluate our tool in a conversational environment. The scene contains six different characters, forming three pairs of conversation. Each pair of characters faces each other while speaking.

4.3 Procedure

The user study took place in Zoom meetings and lasted about 1 hour each. A researcher ran the Unity project on their own device and shared the game window with the participants via Zoom. The researcher demonstrated the system's features in a demo scene. Then the participants used the Zoom remote control to explore the scenes while using the system. One blind participant, unable to access the Zoom remote control, verbally provided commands for a researcher to put into the program and heard system response in real time. Each scene lasted 10 minutes, and the order of scenes was randomly generated to reduce bias.

Participants were instructed to open our built-in help menu for reference when forming their prompts, and they were encouraged to freely explore the scenes and try different prompts on their own. After exploring these two scenes, the participants were asked to complete a post-study survey, where they rated: (1) confidence in enhancing accessibility, (2) system intuitiveness, (3) system usefulness, and (4) usability through the System Usability Scale (SUS) survey [10]. We used the 5-point Likert Scale as a rating scale to assess participants' opinions through questions (1), (2), and (3).

After collecting the survey, we held an interview and asked participants some questions about their experience including (i) system usability, (ii) accessibility needs, (iii) application of the system, and (iv) suggestions for improvement.

4.4 Data Analysis

We used descriptive statistics to summarize the survey data, which included calculating the mean and standard deviation for the ratings. For interview responses, we retained the transcripts and used them to conduct a thematic analysis using Braun and Clarke's six-phased approach [9, 14]. The final codebook contains 56 codes, 16 second-level themes, and seven first-level themes, which we used to form a narrative and produce our findings.

4.5 Findings

Our preliminary results showed insights into system usability, usage patterns, and integration of the system into applications.

4.5.1 System Usability. On average, the participants found the system intuitive to use (*mean*=3.83, *SD*=1.17) and were moderately confident in using the system to enhance accessibility (*mean*=3.67, *SD*=1.03). The users found the query feature to be very useful (*mean*=4.83, *SD*=0.41), followed closely by modification (*mean*=4.50, *SD*=0.84). Our SUS score was 72.92, suggesting reasonable usability with room for improvement. Many participants found the system innovative and exciting. As P6 pointed out, he has not seen similar technology before in existing media, and he was "*really excited to see how this AI could [help me] understand the sound*". Participants noted the system's robustness against typos and praised its conversational interface for being more immersive than traditional accessibility tools, such as captions. However, users also identified challenges related to AI characteristics, such as randomness, hallucinations, and inability to identify ambiguity in prompts. P2 suggested that the LLM agent should offer feedback upon failure and ask follow-up questions in cases of uncertainty.

4.5.2 Usage Patterns. Our diverse group of participants demonstrated a wide range of system usage. Below, we summarize key usage patterns and usability challenges that arose.

The most popular prompt for visual accessibility is obtaining scene descriptions. All BLV participants valued the system's ability to provide a "general picture of the scene". The main issue with the scene descriptions was that they lacked awareness of the user's perspective, like where they are facing, and used the less intuitive 3D coordinates instead. Another issue was their tendency to mechanically list items in the scene rather than providing organic, context-driven descriptions. Participants suggested enhancing descriptions by improving spatial understanding, enabling verbosity adjustment, and adding contexts of the game. In terms of modification, low-vision users considered color and brightness adjustments good strategies but found the default modifications insufficient. Interestingly, the totally blind user mentioned that although she "*didn't care [...] about color*", the system could be very useful if the game objective required color understanding (e.g. collecting all the red objects in the room).

All hard-of-hearing participants found the tool's ability to modify sounds useful for sound accessibility, particularly the ability to selectively mute sound source(s), which could help "*concentrate on a single conversation*" or "*remove distraction*" (P2). DHH participants commented that using prompts could be more efficient than manually adjusting sound parameters. The system's assistance in understanding both dialogue and non-speech sounds is also appreciated. Participants explored innovative prompts to improve sound accessibility, like prompting a speaking NPC to follow the player, generating captions, using familiar sound effects, adding realistic lip movements, or providing AI-generated transcript summaries next to NPCs. Although these prompts are not fully supported yet, they highlight directions for future work.

Many participants suggested that the system should provide better assistance to focus on points of interest, such as items selected, modified, or currently described. These assistance methods could include localized sound notification, direction visualization on screen, or adjusting player viewpoint.

An interesting perspective is the multi-modality of the tool. Users with certain disabilities might limit themselves to certain categories of the tool, like P1 said "*[as a DHH user] I don't really use these [visual accessibility] features normally*". However, it is shown that query and modification in both visual and audio dimensions could be helpful for both groups. For example, a DHH user prompted "*move me to the person talking about augmented reality*", and a BLV user asked the system to "*make the telephone ringing louder*". These multi-modal and cross-modal usage of the system shows promise in integrating and utilizing information from different channels.

4.5.3 Integration into Applications. The participants commented on how the system gives them control over the scene. Although these controls could improve accessibility, they also discussed the possibility of the system "breaking the game", resonating with previous work [25]. To counter this, P1 suggested limiting the

toolkit's scope to a restricted set of accessibility improvements, rather than making it too general. To enhance system integration, users suggested seamless incorporation of game UI, such as using existing icons, and aligning modifications with game mechanics, like combining the "pick up" prompt with the "grab" game action.

5 Conclusion and Future Work

In this paper, we proposed SceneGenA11y, a runtime generative tool to improve accessibility in 3D virtual scenes. We evaluated the system with BLV and DHH users. While our studies are preliminary, they yielded promising results while highlighting potential areas of improvement. Based on our findings, we plan to improve the system and expand our work in the following areas: expand the participant sample; fine-tune the LLM agent to further understand spatial relationships; enhance multi-modal and accessible feedback; and redesign the system to proactively respond to the user's exploration. We will also conduct comparison studies with state-of-the-art Generative AI tools, like Google Multimodal Live API [77, 78]. More generally, our future work will be guided by the question: How to build long-term trust between users and the system? We believe this research will shed light on an innovative and exciting runtime-generative mechanism for virtual 3D scene accessibility.

References

[1] Rudaiba Adnin and Maitraye Das. 2024. "I look at it as the king of knowledge": How Blind People Use and Understand Generative AI Tools. In *Proceedings of the 26th International ACM SIGACCESS Conference on Computers and Accessibility* (*ASSETS '24*), October 27, 2024. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3663548.3675631

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikołaj Bińkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. 2022. Flamingo: a Visual Language Model for Few-Shot Learning. *Advances in Neural Information Processing Systems* 35, (December 2022), 23716–23736.

[3] Rahaf Alharbi, Pa Lor, Jaylin Herskovitz, Sarita Schoenebeck, and Robin N. Brewer. 2024. Misfitting With AI: How Blind People Verify and Contest AI Errors. In *Proceedings of the 26th International ACM SIGACCESS Conference on Computers and Accessibility* (*ASSETS '24*), October 27, 2024. Association for Computing Machinery, New York, NY, USA, 1–17. https://doi.org/10.1145/3663548.3675659

[4] Wajdi Aljedaani, Abdulrahman Habib, Ahmed Aljohani, Marcelo Eler, and Yunhe Feng. 2024. Does ChatGPT Generate Accessible Code? Investigating Accessibility Challenges in LLM-Generated Source Code. In *Proceedings of the 21st International Web for All Conference* (*W4A '24*), October 22, 2024. Association for Computing Machinery, New York, NY, USA, 165–176. https://doi.org/10.1145/3677846.3677854

[5] Ferran Argelaguet and Carlos Andujar. 2013. A survey of 3D object selection techniques for virtual environments. *Computers & Graphics* 37, 3 (May 2013), 121–136. https://doi.org/10.1016/j.cag.2012.12.003

[6] Ben Bayliss. 2022. Fortnite Accessibility — Menu Deep Dive. *Can I Play That?* Retrieved January 24, 2025 from https://caniplaythat.com/2022/04/26/fortnite-accessibility-menu-deep-dive/

[7] Doug A. Bowman, Chris North, Jian Chen, Nicholas F. Polys, Pardha S. Pyla, and Umur Yilmaz. 2003. Information-rich virtual environments: theory, tools, and research agenda. In *Proceedings of the ACM symposium on Virtual reality software and technology* (*VRST '03*), October 01, 2003. Association for Computing Machinery, New York, NY, USA, 81–90. https://doi.org/10.1145/1008653.1008669

[8] Danielle Bragg, Nicholas Huynh, and Richard E. Ladner. 2016. A Personalizable Mobile Sound Detector App Design for Deaf and Hard-of-Hearing Users. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility* (*ASSETS '16*), October 23, 2016. Association for Computing Machinery, New York, NY, USA, 3–13. https://doi.org/10.1145/2982142.2982171

[9] Virginia Braun, Victoria Clarke, Nikki Hayfield, Louise Davey, and Elizabeth Jenkinson. 2022. Doing Reflexive Thematic Analysis. In *Supporting Research in Counselling and Psychotherapy: Qualitative, Quantitative, and Mixed Methods Research*, Sofie Bager-Charleson and Alistair McBeath (eds.). Springer International Publishing, Cham, 19–38. https://doi.org/10.1007/978-3-031-13942-0_2

[10] john Brooke. 1996. SUS: A "Quick and Dirty" Usability Scale. In *Usability Evaluation In Industry*. CRC Press.

[11] Xinyun Cao and Dhruv Jain. 2024. SoundModVR: Sound Modifications in Virtual Reality to Support People who are Deaf and Hard of Hearing. In *Proceedings of the 26th International ACM SIGACCESS Conference on Computers and Accessibility* (*ASSETS '24*), October 27, 2024. Association for Computing Machinery, New York, NY, USA, 1–15. https://doi.org/10.1145/3663548.3675653

[12] Ruei-Che Chang, Chia-Sheng Hung, Bing-Yu Chen, Dhruv Jain, and Anhong Guo. 2024. SoundShift: Exploring Sound Manipulations for Accessible Mixed-Reality Awareness. In *Proceedings of the 2024 ACM Designing Interactive Systems Conference* (*DIS '24*), July 01, 2024. Association for Computing Machinery, New York, NY, USA, 116–132. https://doi.org/10.1145/3643834.3661556

[13] Ruei-Che Chang, Yuxuan Liu, and Anhong Guo. 2024. WorldScribe: Towards Context-Aware Live Visual Descriptions. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (*UIST '24*), October 11, 2024. Association for Computing Machinery, New York, NY, USA, 1–18. https://doi.org/10.1145/3654777.3676375

[14] Victoria Clarke and Virginia Braun. 2017. Thematic analysis. *The Journal of Positive Psychology* 12, 3 (May 2017), 297–298. https://doi.org/10.1080/17439760.2016.1262613

[15] Fernanda De La Torre, Cathy Mengying Fang, Han Huang, Andrzej Banburski-Fahey, Judith Amores Fernandez, and Jaron Lanier. 2024. LLMR: Real-time Prompting of Interactive Worlds using Large Language Models. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (*CHI '24*), May 11, 2024. Association for Computing Machinery, New York, NY, USA, 1–22. https://doi.org/10.1145/3613904.3642579

[16] Nadeen Fathallah, Monika Bhole, and Steffen Staab. 2024. Empowering the Deaf and Hard of Hearing Community: Enhancing Video Captions Using Large Language Models. *arXiv.org*. Retrieved January 24, 2025 from https://arxiv.org/abs/2412.00342v1

[17] Krzysztof Z. Gajos, Amy Hurst, and Leah Findlater. 2012. Personalized dynamic accessibility. *interactions* 19, 2 (March 2012), 69–73. https://doi.org/10.1145/2090150.2090167

[18] Kate S Glazko, Momona Yamagami, Aashaka Desai, Kelly Avery Mack, Venkatesh Potluri, Xuhai Xu, and Jennifer Mankoff. 2023. An Autoethnographic Case Study of Generative Artificial Intelligence's Utility for Accessibility. In *Proceedings of the 25th International ACM SIGACCESS Conference on Computers and Accessibility* (*ASSETS '23*), October 22, 2023. Association for Computing Machinery, New York, NY, USA, 1–8. https://doi.org/10.1145/3597638.3614548

[19] Joshua Gorniak, Yoon Kim, Donglai Wei, and Nam Wook Kim. 2024. VizAbility: Enhancing Chart Accessibility with LLM-based Conversational Interaction. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (*UIST '24*), October 11, 2024. Association for Computing Machinery, New York, NY, USA, 1–19. https://doi.org/10.1145/3654777.3676414

[20] Harun. XRA'S DEVELOPERS GUIDE, CHAPTER THREE: Accessibility & Inclusive Design in Immersive Experiences. *XR Association*. Retrieved March 7, 2025 from https://xra.org/research/xra-developers-guide-accessibility-and-inclusive-design/

[21] Ziniu Hu, Ahmet Iscen, Aashi Jain, Thomas Kipf, Yisong Yue, David A. Ross, Cordelia Schmid, and Alireza Fathi. 2024. SceneCraft: An LLM Agent for Synthesizing 3D Scenes as Blender Code. June 06, 2024. . Retrieved January 13, 2025 from https://openreview.net/forum?id$=$gAyzjHw2ml

[22] L. L. Hunter, R. H. Margolis, J. R. Rykken, C. T. Le, K. A. Daly, and G. S. Giebink. 1996. High frequency hearing loss associated with otitis media. *Ear Hear* 17, 1 (February 1996), 1–11. https://doi.org/10.1097/00003446-199602000-00001

[23] Dhruv Jain, Sasa Junuzovic, Eyal Ofek, Mike Sinclair, John R. Porter, Chris Yoon, Swetha Machanavajhala, and Meredith Ringel Morris. 2021. Towards Sound Accessibility in Virtual Reality. In *Proceedings of the 2021 International Conference on Multimodal Interaction* (*ICMI '21*), October 18, 2021. Association for Computing Machinery, New York, NY, USA, 80–91. https://doi.org/10.1145/3462244.3479946

[24] G. Jansson, H. Petrie, C. Colwell, D. Kornbrot, J. Fänger, H. König, K. Billberger, A. Hardwick, and S. Furner. 1999. Haptic Virtual Environments for Blind People: Exploratory Experiments with Two Devices. *International Journal of Virtual Reality* 4, 1 (January 1999), 8–17. https://doi.org/10.20870/IJVR.1999.4.1.2663

[25] Nicholas Jennings, Han Wang, Isabel Li, James Smith, and Bjoern Hartmann. 2024. What's the Game, then? Opportunities and Challenges for Runtime Behavior Generation. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (*UIST '24*), October 11, 2024. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3654777.3676358

[26] Tiger F. Ji, Yaxin Hu, Yu Huang, Ruofei Du, and Yuhang Zhao. 2023. A Preliminary Interview: Understanding XR Developers' Needs towards Open-Source Accessibility Support. In *2023 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, March 2023. 493–496. https://doi.org/10.1109/VRW58643.2023.00107

[27] Peter E Jones. 1999. Three-dimensional input device with six degrees of freedom. *Mechatronics* 9, 7 (October 1999), 717–729. https://doi.org/10.1016/S0957-4158(99)00032-X

[28] kevinasg. 2023. Xbox Accessibility Guidelines - Microsoft Game Dev. Retrieved March 7, 2025 from https://learn.microsoft.com/en-us/gaming/accessibility/guidelines

[29] Satwik Ram Kodandaram, Utku Uckun, Xiaojun Bi, IV Ramakrishnan, and Vikas Ashok. 2024. Enabling Uniform Computer Interaction Experience for Blind Users through Large Language Models. In *Proceedings of the 26th International ACM SIGACCESS Conference on Computers and Accessibility* (*ASSETS '24*), October 27, 2024. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3663548.3675605

[30] Zhifeng Kong, Arushi Goel, Rohan Badlani, Wei Ping, Rafael Valle, and Bryan Catanzaro. 2024. Audio Flamingo: A Novel Audio Language Model with Few-Shot Learning and Dialogue Abilities. https://doi.org/10.48550/arXiv.2402.01831

[31] A. Lecuyer, P. Mobuchon, C. Megard, J. Perret, C. Andriot, and J.-P. Colinot. 2003. HOMERE: a multimodal system for visually impaired people to explore virtual environments. In *IEEE Virtual Reality, 2003. Proceedings.*, March 2003. 251–258. https://doi.org/10.1109/VR.2003.1191147

[32] Ziming Li, Shannon Connell, Wendy Dannels, and Roshan Peiris. 2022. SoundVizVR: Sound Indicators for Accessible Sounds in Virtual Reality for Deaf or Hard-of-Hearing Users. In *Proceedings of the 24th International ACM SIGACCESS Conference on Computers and Accessibility* (*ASSETS '22*), October 22, 2022. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3517428.3544817

[33] Vardit Lichtenstein and David R. Stapells. 1996. Frequency-specific identification of hearing loss using transient-evoked otoacoustic emissions to clicks and tones. *Hearing Research* 98, 1 (September 1996), 125–136. https://doi.org/10.1016/0378-5955(96)00084-6

[34] Cher P. Lim, Darren Nonis, and John Hedberg. 2006. Gaming in a 3D multiuser virtual environment: engaging students in Science lessons. *British Journal of Educational Technology* 37, 2 (2006), 211–231. https://doi.org/10.1111/j.1467-8535.2006.00531.x

[35] Shachar Maidenbaum, Shelly Levy-Tzedek, Daniel-Robert Chebat, and Amir Amedi. 2013. Increasing Accessibility to the Blind of Virtual Environments, Using a Virtual Mobility Aid Based On the "EyeCane": Feasibility Study. *PLOS ONE* 8, 8 (August 2013), e72555. https://doi.org/10.1371/journal.pone.0072555

[36] Mohammadreza Mirzaei, Peter Kán, and Hannes Kaufmann. 2021. Multi-modal Spatial Object Localization in Virtual Reality for Deaf and Hard-of-Hearing People. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*, March 2021. 588–596. https://doi.org/10.1109/VR50410.2021.00084

[37] Mario Montagud, Issac Fraile, Juan A. Nuñez, and Sergi Fernández. 2018. ImAc: Enabling Immersive, Accessible and Personalized Media Experiences. In *Proceedings of the 2018 ACM International Conference on Interactive Experiences for TV and Online Video* (*TVX '18*), June 25, 2018. Association for Computing Machinery, New York, NY, USA, 245–250. https://doi.org/10.1145/3210825.3213570

[38] Peya Mowar, Yi-Hao Peng, Aaron Steinfeld, and Jeffrey P Bigham. 2024. Tab to Autocomplete: The Effects of AI Coding Assistants on Web Accessibility. In *Proceedings of the 26th International ACM SIGACCESS Conference on Computers and Accessibility* (*ASSETS '24*), October 27, 2024. Association for Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/3663548.3688513

[39] NicholasJJ. 2024. NicholasJJ/GROMIT. Retrieved January 24, 2025 from https://github.com/NicholasJJ/GROMIT

[40] Achraf Othman, Amira Dhouib, and Aljazi Nasser Al Jabor. 2023. Fostering websites accessibility: A case study on the use of the Large Language Models ChatGPT for automatic remediation. In *Proceedings of the 16th International Conference on PErvasive Technologies Related to Assistive Environments* (*PETRA '23*), August 10, 2023. Association for Computing Machinery, New York, NY, USA, 707–713. https://doi.org/10.1145/3594806.3596542

[41] Christine Rigden. 1999. 'The Eye of the Beholder'— Designing for Colour-Blind Users. *HUMAN FACTORS* 17, (1999).

[42] Gary S. Rubin, Mary Feely, Sylvie Perera, Katherin Ekstrom, and Elizabeth Williamson. 2006. The effect of font and line width on reading speed in people with mild to moderate vision loss. *Ophthalmic and Physiological Optics* 26, 6 (2006), 545–554. https://doi.org/10.1111/j.1475-1313.2006.00409.x

[43] JAIME SÁNCHEZ and MAURICIO LUMBRERAS. 1999. Virtual Environment Interaction Through 3D Audio by Blind Children. *CyberPsychology & Behavior* 2, 2 (April 1999), 101–111. https://doi.org/10.1089/cpb.1999.2.101

[44] S K Semwal. MoVE: Mobiltiy Training in Haptic Virtual Environment.

[45] JooYoung Seo, Sanchita S. Kamath, Aziz Zeidieh, Saairam Venkatesh, and Sean McCurry. 2024. MAIDR Meets AI: Exploring Multimodal LLM-Based Data Visualization Interpretation by and with Blind and Low-Vision Users. In *Proceedings of the 26th International ACM SIGACCESS Conference on Computers and Accessibility* (*ASSETS '24*), October 27, 2024. Association for Computing Machinery, New York, NY, USA, 1–31. https://doi.org/10.1145/3663548.3675660

[46] Abigale Stangl, Nitin Verma, Kenneth R. Fleischmann, Meredith Ringel Morris, and Danna Gurari. 2021. Going Beyond One-Size-Fits-All Image Descriptions to Satisfy the Information Wants of People Who are Blind or Have Low Vision. In *Proceedings of the 23rd International ACM SIGACCESS Conference on Computers and Accessibility* (*ASSETS '21*), October 17, 2021. Association for Computing Machinery, New York, NY, USA, 1–15. https://doi.org/10.1145/3441852.3471233

[47] Richard J. Stubbs and Quentin Summerfield. 2005. Separation of simultaneous voices. *The Journal of the Acoustical Society of America* 81, S1 (August 2005), S60. https://doi.org/10.1121/1.2024313

[48] Robert H. Tai, Lillian R. Bentley, Xin Xia, Jason M. Sitt, Sarah C. Fankhauser, Ana M. Chicas-Mosier, and Barnas M. Monteith. 2023. Use of Large Language Models to Aid Analysis of Textual Data. 2023.07.17.549361. https://doi.org/10.1101/2023.07.17.549361

[49] Dimitrios Tzovaras, Konstantinos Moustakas, Georgios Nikolakis, and Michael G. Strintzis. 2009. Interactive mixed reality white cane simulation for the training of the blind and the visually impaired. *Pers Ubiquit Comput* 13, 1 (January 2009), 51–58. https://doi.org/10.1007/s00779-007-0171-2

[50] Jacob T. Urbina, Peter D. Vu, and Michael V. Nguyen. 2025. Disability Ethics and Education in the Age of Artificial Intelligence: Identifying Ability Bias in ChatGPT and Gemini. *Archives of Physical Medicine and Rehabilitation* 106, 1 (January 2025), 14–19. https://doi.org/10.1016/j.apmr.2024.08.014

[51] Songül Atasavun Uysa and Tülin Düger. 2012. Writing and Reading Training Effects on Font Type and Size Preferences by Students with Low Vision. *Percept Mot Skills* 114, 3 (June 2012), 837–846. https://doi.org/10.2466/15.10.11.24.PMS.114.3.837-846

[52] Alistair Wong. 2019. Spice & Wolf VR Leaves You Wanting For More Of Lawrence And Holo's Sweet Interactions. *Siliconera*. Retrieved January 24, 2025 from https://www.siliconera.com/spice-wolf-vr-leaves-you-wanting-for-more-of-lawrence-and-holos-sweet-interactions/

[53] Liang-Yuan Wu, Andrea Kleiver, and Dhruv Jain. 2024. CARTGPT: Improving CART Captioning using Large Language Models. In *Proceedings of the 26th International ACM SIGACCESS Conference on Computers and Accessibility* (*ASSETS '24*), October 27, 2024. Association for Computing Machinery, New York, NY, USA, 1–5. https://doi.org/10.1145/3663548.3688494

[54] Lee H. Wurm, Gordon E. Legge, Lisa M. Isenberg, and Andrew Luebker. 1993. Color improves object recognition in normal and low vision. *Journal of Experimental Psychology: Human Perception and Performance* 19, 4 (1993), 899–911. https://doi.org/10.1037/0096-1523.19.4.899

[55] Yue Yang, Fan-Yun Sun, Luca Weihs, Eli VanderBilt, Alvaro Herrasti, Winson Han, Jiajun Wu, Nick Haber, Ranjay Krishna, Lingjie Liu, Chris Callison-Burch, Mark Yatskar, Aniruddha Kembhavi, and Christopher Clark. 2024. Holodeck: Language Guided Generation of 3D Embodied AI Environments. 2024. 16227–16237. Retrieved January 24, 2025 from https://openaccess.thecvf.com/content/CVPR2024/html/Yang_Holodeck_Language_Guided_Generation_of_3D_Embodied_AI_Environments_CVPR_2024_paper.html

[56] Zhe-Xin Zhang and Yoichi Ochiai. 2024. A Design of Interface for Visual-Impaired People to Access Visual Information from Images Featuring Large Language Models and Visual Language Models. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems* (*CHI EA '24*), May 11, 2024. Association for Computing Machinery, New York, NY, USA, 1–4. https://doi.org/10.1145/3613905.3648648

[57] Ziyao Zhang, Yanlin Wang, Chong Wang, Jiachi Chen, and Zibin Zheng. 2024. LLM Hallucinations in Practical Code Generation: Phenomena, Mechanism, and Mitigation. *arXiv.org*. Retrieved January 24, 2025 from https://arxiv.org/abs/2409.20550v2

[58] Yuhang Zhao, Cynthia L. Bennett, Hrvoje Benko, Edward Cutrell, Christian Holz, Meredith Ringel Morris, and Mike Sinclair. 2018. 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*), April 19, 2018. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3173574.3173690

[59] 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* (*CHI '19*), May 02, 2019. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3290605.3300341

[60] 2018. The Persistence for PS VR Gets Huge Free Update October 18. *PlayStation.Blog*. Retrieved January 24, 2025 from https://blog.playstation.com/2018/10/11/the-persistence-for-ps-vr-gets-huge-free-update-october-18/

[61] 2021. Minecraft Accessibility. *Minecraft.net*. Retrieved February 18, 2024 from https://www.minecraft.net/en-us/accessibility

[62] 2024. (PDF) How do blind people perceive sound and soundscape? *ResearchGate* (October 2024). Retrieved January 22, 2025 from https://www.researchgate.net/publication/282273661_How_do_blind_people_perceive_sound_and_soundscape

[63] 2024. (PDF) The Effect Of 3D Audio And Other Audio Techniques On Virtual Reality Experience. *ResearchGate* (October 2024). https://doi.org/10.3233/978-1-61499-595-1-44

[64] WHP330.pdf. Retrieved March 6, 2025 from https://downloads.bbc.co.uk/rd/pubs/whp/whp-pdf-files/WHP330.pdf

[65] A Training System of Orientation and Mobility for Blind People Using Acoustic Virtual Reality | IEEE Journals & Magazine | IEEE Xplore. Retrieved January 24, 2025 from https://ieeexplore.ieee.org/abstract/document/5559478

[66] Seeing the world by hearing: Virtual Acoustic Space (VAS) a new space perception system for blind people. | IEEE Conference Publication | IEEE Xplore. Retrieved January 24, 2025 from https://ieeexplore.ieee.org/document/1684482
[67] EarVR: Using Ear Haptics in Virtual Reality for Deaf and Hard-of-Hearing People | IEEE Journals & Magazine | IEEE Xplore. Retrieved January 24, 2025 from https://ieeexplore.ieee.org/abstract/document/8998298
[68] Enable Blind Users' Experience in 3D Virtual Environments: The Scene Weaver Prototype | Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems. Retrieved January 24, 2025 from https://dl.acm.org/doi/full/10.1145/3544549.3583909
[69] Exploration of architectural spaces by blind people using auditory virtual reality for the construction of spatial knowledge - ScienceDirect. Retrieved January 24, 2025 from https://www.sciencedirect.com/science/article/pii/S1071581913002036
[70] Toward accessible 3D virtual environments for the blind and visually impaired | Proceedings of the 3rd international conference on Digital Interactive Media in Entertainment and Arts. Retrieved January 24, 2025 from https://dl.acm.org/doi/abs/10.1145/1413634.1413663
[71] The Last of Us Part II - Accessibility. *PlayStation*. Retrieved January 24, 2025 from https://www.playstation.com/en-us/games/the-last-of-us-part-ii/accessibility/
[72] ChatGPT can now see, hear, and speak. Retrieved January 24, 2025 from https://openai.com/index/chatgpt-can-now-see-hear-and-speak/
[73] Introducing: Be My AI. Retrieved January 24, 2025 from https://www.bemyeyes.com/blog/introducing-be-my-ai
[74] Web Content Accessibility Guidelines (WCAG) 2.0. Retrieved January 24, 2025 from https://www.w3.org/TR/WCAG20/
[75] VRCopilot: Authoring 3D Layouts with Generative AI Models in VR | Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology. Retrieved January 24, 2025 from https://dl.acm.org/doi/abs/10.1145/3654777.3676451
[76] VoiceAttack - Voice Recognition for your Games and Apps. Retrieved January 22, 2025 from https://voiceattack.com/
[77] GitHub - google-gemini/multimodal-live-api-web-console: A react-based starter app for using the Multimodal Live API over websockets with Gemini. Retrieved March 7, 2025 from https://github.com/google-gemini/multimodal-live-api-web-console
[78] Multimodal Live API | Gemini API. *Google AI for Developers*. Retrieved March 7, 2025 from https://ai.google.dev/gemini-api/docs/multimodal-live

A APPENDICES

A Static Priming Message

A11y Prime Prompt

To highlight an object, you can use the GPT Indicator material.

To highlight an object without a Renderer, you create a transparent sphere with GPT Indicator material at its location for 5 seconds.

To select an object, you can create a transparent sphere with GPT Indicator material at its location for 5 seconds.

The following is a section about colors:

The HEX code represent the color. When asked about the color of an object, answer with natural language color instead of HEX code.

Red-green color blindness is a type of color vision deficiency that makes it difficult to distinguish between shades of red and green.

To make a scene more accessible for someone with red green color blindness, you should change the color palette. To make a palette for red green colorblind, avoid combining red and green. Also, make sure the new color created are not the same or similar as the other colors in the surroundings.

The following is a section about simplifying material or texture of an object:

When asked to simplify material or texture of an object, create a new material that is closest to the object's original color and assign this new material to the object. If the original color is not provided, use the best guess given the object name.

To change the color of an object, first simplify the texture and then change the color.

The following is a section about spatial relationship between objects:

When "me, I, my" is referred, it means the player.

When asked about location of objects, answer the object's location relative to the player's location.

When asked about the location of one object relative to another, respond by the distance calculated using euclidean distance between the center of the two objects. Be as presice as possible. Also answer how far the item is to the player in common sense, like "the object is close to you" or "the object is far away from you".

When asked about size of an item, each unit is a meter. Answer how big an object is based on the size in meters.

When asked about how big is a text, answer the font size of the text.

To add light to an area, create a Sphere game object in the area and add a point light to the sphere.

To change the range of a sound source, change the max distance.

When asked to describe the scene or what are in the scene, describe it briefly, group similar objects together instead of listing all items.

If the request is general or incomplete, please ask follow-up questions for precise details and contexts, and leave the 'code' field null.

If the request says it's not working, please ask follow-up questions to clarify what's happening and suggest users to refine their request. If it's still not working, apologize to users and ask them to try another task.

If the request is out of your capability, tell users that the request is out of scope. The types of requests that cannot be achieved include: make zoom/magnifier, edge enhancement, color change on textured materials, object deletion.

B Dynamic Priming

```
public string GetDescription() {
        string temp = description;
        Color? color = FindColor();
        if (color != null){
            temp += \$" The color of this
item has the HEX code {ColorUtility.ToHtmlStringRGB(color.Value)}.
        }
        TextMeshProUGUI text = GetComponent<TextMeshProUGUI>();
        if (text != null){
            temp += \$" The font size of the
            text on this item is {text.fontSize}";
        }
        Light light = GetComponent<Light>();
        if (light != null){
            temp += \$" The intensity of the light
source on this item is {light.intensity}";
        }
        AudioSource audioSource = GetComponent<AudioSource>();
        if (audioSource != null){
            temp += \$" For the audio source on this item,";
            if (audioSource.mute){
                temp += \$" it is muted,";
            } else{
                temp += \$" it is not muted,";
```

```
    }
    temp += \$" the volume is {audioSource.volume}";
    temp += \$" the pitch is {audioSource.pitch}";
    temp += \$" the range is {audioSource.maxDistance}";
}
TextMeshProUGUI textElement = GetComponent<TextMeshProUGUI>();
if (textElement != null){
    temp += \$" the text on this item says:
     " + textElement.text;
}
return temp;
```