SPATIALIZING SCREEN READERS: EXTENDING VOICEOVER VIA HEAD-TRACKED BINAURAL SYNTHESIS FOR USER INTERFACE ACCESSIBILITY

Giuseppe Bergamino , Michael Fioretti , Leonardo Gabrielli and Stefano Squartini

Dept. of Information Engineering
Università Politecnica delle Marche
Ancona, IT
<g.bergamino,m.fioretti>@pm.univpm.it, <l.gabrielli,s.squartini>@staff.univpm.it

ABSTRACT

Traditional screen-based graphical user interfaces (GUIs) pose significant accessibility challenges for visually impaired users. This paper demonstrates how existing GUI elements can be translated into an interactive auditory domain using high-order Ambisonics and inertial sensor-based head tracking, culminating in a real-time binaural rendering over headphones. The proposed system is designed to spatialize the auditory output from VoiceOver, the built-in macOS screen reader, aiming to foster clearer mental mapping and enhanced navigability. A between-groups experiment was conducted to compare standard VoiceOver with the proposed spatialized version. Non visually-impaired participants (n = 32), with no visual access to the test interface, completed a list-based exploration and then attempted to reconstruct the UI solely from auditory cues. Experimental results indicate that the head-tracked group achieved a slightly higher accuracy in reconstructing the interface, while user experience assessments showed no significant differences in self-reported workload or usability. These findings suggest that potential benefits may come from the integration of head-tracked binaural audio into mainstream screen-reader workflows, but future investigations involving blind and low-vision users are needed. Although the experimental testbed uses a generic desktop app, our ultimate goal is to tackle the complex visual layouts of music-production software, where an head-tracked audio approach could benefit visually impaired producers and musicians navigating plug-in controls.

1. INTRODUCTION

For decades, human–computer interaction (HCI) has primarily relied on visual output, ranging from text-based terminals to sophisticated graphical user interfaces (GUIs). Although this evolution has broadened functionality and user engagement, it has simultaneously introduced accessibility barriers for those with visual impairments. According to the World Health Organization, at least 2.2 billion people worldwide live with some form of vision impairment [1]. Traditional screen-based GUIs thus create a substantial usability gap for this large user group. In parallel, the HCI community has increasingly explored multimodal interfaces, which incorporate additional sensory channels, such as auditory or haptic, to offer richer and more adaptable interaction experiences [2]. By reducing the heavy reliance on vision, multimodality can help address the needs of visually impaired users. However, many existing auditory user interfaces (AUIs) –ranging from simple text-to-speech readers to more advanced screen readers – still impose a high cognitive and memory load. As noted by Edwards in his seminal study on auditory interfaces for visually disabled users [3], the linear and sequential nature of audio feedback often forces users to memorize large amounts of information, especially when navigating complex interfaces. To mitigate this memory overhead, our work proposes spatializing the audio output of standard screen readers. In particular, we extend Apple VoiceOver with a high-order Ambisonics pipeline and real-time head tracking, enabling users to explore and localize interface elements in a spherical auditory space. By mapping UI components to distinct azimuth and elevation positions around the listener, our goal is to foster a more intuitive and persistent *mental map* of the interface layout.

The remainder of this paper is organized as follows. We first review the key related works on non-visual access and 3D audio interfaces, highlighting the the gaps that motivate our system design (Section 1.1). We then describe our spatialized VoiceOver implementation (Section 2), outline the experimental design (Section 3), and present an evaluation comparing our approach to conventional VoiceOver (Section 4). Finally, Section 5 discusses the conclusions and future directions.

1.1. Related works

Auditory user interfaces (AUIs) have historically evolved adding audio cues on top of visual interfaces, relying on paradigms such as auditory icons [4], earcons [5], or a mix of both [6]. Various studies have explored the effectiveness of spatialized auditory information, including the use of head tracking [7] or augmented reality frameworks [8]. While these sonification approaches can enrich the user experience, they may require a certain degree of *musical* focus to interpret changes in pitch or audio icons, potentially adding extra cognitive demands on the user. In contrast, text-to-speech (TTS) coupled with a screen reader typically builds on users' everyday familiarity with spoken language. Since a large segment of the population is accustomed to listening to voice-based content (e.g., podcasts, virtual assistants), it seems plausible that voiced screen-reader output might impose less additional cognitive load compared to purely sonified elements.

A screen reader is an assistive technology that conveys digital text or images as synthesized speech or braille output. Screen readers are available as standalone third-party software or can be built-in features of desktop and mobile operating systems. They enable a user to navigate content linearly using the platform's native input methods, such as touch gestures on smartphones or keyboard input on desktop. By reading aloud on-screen elements and providing audio cues for focus changes, screen readers offer fun-

damental accessibility for people with little or no residual vision. Several works have investigated the integration of speech synthesis with screen readers—highlighting both opportunities (e.g., clarity of verbal feedback) and challenges (e.g., linear navigation, verbosity)—and explored 3D positioning of synthesized speech as a means to further enhance non-visual access.

For instance, Crispien et al. [9] proposed a hardware-based solution using HRTFs and head tracking. This system placed textual elements within a virtual acoustic free-field, allowing users to discern location via distance-based filtering. Although innovative and pioneering, the approach required specialized hardware and offered limited resolution compared to modern high-order Ambisonics or software-driven pipelines.

Similarly, Goose et al. [10] presented a 3D audio-only web browser, using spatialization to convey hypermedia document structure. The system mapped HTML elements onto a virtual sound field, relying on speech synthesis to read each segment of the page. Despite introducing interesting positional cues along the x-axis to indicate a user's position in the document, it primarily targeted web navigation rather than general desktop or GUI-based applications. Moreover, no formal usability test was reported, leaving open questions about cognitive load and user performance in real-world scenarios.

Sodnik et al. [11] proposed an enhanced synthesized text reader capable of placing multiple voices at distinct 3D positions, aiming to facilitate e-book reading for visually impaired users. By embedding metadata in the text file, different voices (with varied pitch, rate, etc.) could be spatially mapped around the listener, potentially improving engagement and scene comprehension. Although they integrated an external HRIR library to boost localization accuracy, the approach did not employ head tracking, limiting the sense of dynamic spatial exploration.

Meanwhile, Morris et al. [12] focused on enhanced representations of visual content for screen reader users, especially for images on the web. Their system extends the notion of alt text by introducing an interactive design space encompassing multiple properties (e.g., interactivity, representation, personalization). Although one of their prototypes supports a spatial interaction style for images—allowing users to touch different regions of an image on a touchscreen—the emphasis remains on static content (HTML documents) rather than dynamic, OS-level interfaces. Moreover, the spatial aspects are limited to localized image regions, as opposed to a full 3D auditory layout for the entire user interface.

Zong et al. [13] tackled rich screen reader experiences for accessible data visualization, proposing novel design dimensions structure, navigation, and description to adapt visual charts to a screen-reader-friendly format. Through a co-design process, they explored ways to help users conceptualize data spatially, supporting multiple levels of granularity and reducing cognitive overload. Although their work focuses on data visualization in web-based contexts, the underlying principle of providing bounded rooms for navigation resonates with our goal of lowering cognitive load through spatial segmentation. However, Zong et al. primarily address chart structures and targeted static or interactive data exploration, whereas we extend a head-tracked 3D audio environment to a more general UI layout, enabling spatial interaction with all interface elements rather than individual data points in a chart.

Lastly, Chheda-Kothary et al. [14] investigated spatial interactions in desktop screen readers through their custom *SpaceNav* prototype, focusing on web applications that mimic real-world websites. They found that spatial cues could reduce cognitive load and improve orientation for many participants, while some long-time screen reader users found the spatial audio less necessary or even more demanding. The system relies primarily on horizontal placement and uses earcons for vertical cues, but does not employ head tracking or higher-order Ambisonics, thus offering a static rather than fully dynamic interactive experience.

Although Crispien's early work [9] provided a foundational model for head-tracked 3D audio in a screen reader context, and other subsequent studies have explored spatial cues or web-based applications, none fully integrate a high-order Ambisonics pipeline with a modern OS-level screen reader, combined with real-time head tracking for binaural rendering over headphones.

2. SYSTEM DESCRIPTION

Screen readers remain the most common assistive technology for blind and low-vision users to access graphical interfaces [15], with recent WebAIM surveys indicating that JAWS, NVDA, and VoiceOver are among the top three in use [16]. In our approach, UI components recognized by a screen reader are spatialized in a spherical audio field, while a head-tracker attached to the user's headphones provides orientation data to the system (Figure 1). Consequently, each spoken interface element acquires a specific coordinate in the immersive sound field, and when the user physically turns their head, the perceived location of each element shifts accordingly, forming a stable, external acoustic reference.

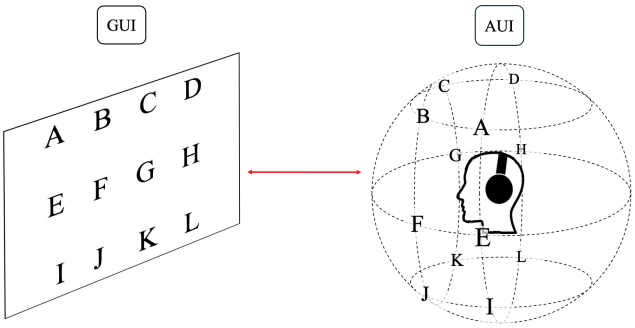

Figure 1: *Graphical description of the concept.*

Rather than converting a fully visual interface into audio, we adopt a *meta user interface* paradigm [17], in which an abstract task model is transformed into both *auditory* and *visual* modalities. This ensures that both representations originate from the same underlying structure, preserving core interactions while allowing modality-specific optimizations. In our design process, we drew on insights from Payne et al. [18], who interviewed blind and low-vision composers, producers, and songwriters about the difficulties of navigating dense music-production software. Many of Payne's participants described relying on sighted assistants to operate GUIs, prompting us to ensure our system supports a shared frame of reference: while sighted collaborators view a 2D grid or windowed interface, non-sighted users encounter an equivalent auditory layout, mapped onto distinct 3D coordinates. This common anchor enables everyone to refer to the same *top-right* or *left column* control, whether perceived visually or via spatialized audio, thereby fostering more direct collaboration among users with different levels of vision.

By distributing interface components in an immersive audio

space, we also aim to reduce the user's memory overhead, allowing them to navigate via dynamic head-tracking and form a clearer mental map of available controls. This approach could benefit not only visually impaired users but also anyone working in visually constrained environments or seeking alternative, more engaging interaction modes. In particular, head-tracked auditory augmented reality improves localization by leveraging proprioceptive feedback through active head movements [19, 20], potentially reinforcing each user's spatial awareness of the interface.

2.1. Implementation

Our prototype system targets macOS VoiceOver [21], chosen for its built-in accessibility features and close integration with the operating system. By intercepting VoiceOver's text-to-speech (TTS) output, we can spatialize every spoken interface element in real time. Specifically, the TTS audio is redirected to a virtual audio device (SoundFlower), which routes the signal into Reaper, a digital audio workstation (DAW) chosen for its flexible routing and multichannel (up to 64) per track capabilities. Within Reaper, each spoken event passes through an Ambisonics processing chain based on the IEM Plug-in Suite [22], where the signal is encoded as 7th order Ambisonics using IEM's *StereoEncoder*. The Ambisonics bus then feeds into IEM's *SceneRotator*, which shifts the entire soundfield inversely to the user's head movements, consistent with data received via OSC from our IMU-based head tracker. Finally, a *BinauralDecoder* converts Ambisonics into a stereo headphone mix, using generic HRTFs (Neumann KU100) to achieve 3D localization.

This design leverages macOS accessibility APIs (VoiceOver) and a conventional DAW pipeline for audio routing, while the front-end UI is developed in JUCE. By offering straightforward integration with the operating system's accessibility APIs, JUCE enables developers to attach labels, roles, and states to each control with minimal code changes, ensuring that the OS can effectively read and announce these elements. Consequently, this approach remains lightweight yet delivers an immersive, stable 3D auditory experience for users, relying entirely on widely available tools.

Figure 2 illustrates the overall system pipeline, summarizing how the user app (Section 2.2) works with VoiceOver, the Ambisonics chain (Section 2.2.2), and the head-tracker integration (Section 2.3).

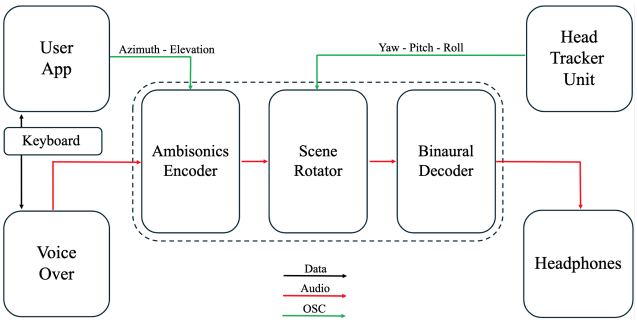

Figure 2: *Block diagram of the system pipeline*

2.1.1. Accessibility functions

Before spatializing the screen reader output, it is crucial to ensure that all controls in the application are properly exposed to the operating system's accessibility framework. As noted in the survey by Payne et al. [18], many musical production UIs lack internal descriptors for each element, hindering assistive technologies. Typically, an operating system sees only an app's window as a single entity, with no insight into the underlying controls, unless developers explicitly expose them via accessibility APIs. Our first step, therefore, is to assign clear labels, roles, and states to each UI component. This allows macOS to discover them at a system-wide level, thereby unlocking not only VoiceOver, but also additional features such as braille displays, head pointers, or alternative interaction methods. In our implementation, we rely on JUCE framework and its `AccessibilityHandler` class [23], which supplies the metadata (e.g. label, description, usage hints) that the OS queries. For instance, a toggle button might be registered with a short label (e.g. Mute), a role (e.g. Button) and its current state (e.g. Off) so that VoiceOver can announce it accurately and track its changes. Similar tagging can apply to sliders, menus, or text fields. By systematically populating these parameters before any visual layout is set, we ensure the application's controls are equally discoverable to all accessibility services, paving the way for subsequent head-tracked, 3D audio rendering.

2.2. User interface design

Our system organizes each application's controls in a hierarchical tree, enabling a clear mapping from high-level (macro) groups down to individual elements and their adjustable parameters. This hierarchical structure offers several advantages:

- It mirrors how large applications (e.g., audio plug-ins) arrange functionality into modules or sections.
- It prepares both the GUI and AUI mappings to maintain consistent, predictable navigation across modalities.
- It speeds up navigation by allowing users to jump among macro groups rather than traversing every single control sequentially.
- It aligns with keyboard navigation practices, benefiting visually impaired users who do not rely on a mouse.

In JUCE, each high-level group is declared as an accessibility node (set as `focusContainer`), with child elements representing the individual controls. By specifying a group name and the total number of items within it, the screen reader can inform the user how many nodes are contained in that group, making navigation more transparent. For demonstration and testing, we developed a simple *Shopping List* desktop application so even participants unfamiliar with music production concepts could evaluate the system. As shown in Table 1, its meta user interface domain consists of five major categories (macro groups) aligned with a typical Italian dinner sequence: *Appetizer*, *First course*, *Second course*, *Side dish*, and *Beverage*. Each group contains four related items (e.g., in the *Beverage* group: red wine, white wine, beer, cola), the amount of which is user selectable and expressed in intuitive units of measurements (e.g. number of bottles and cans). While this example is trivial, it illustrates how the same hierarchical design can scale to full-fledged audio applications, where entire banks of sliders, knobs, and menus form sub-trees of a broader structure.

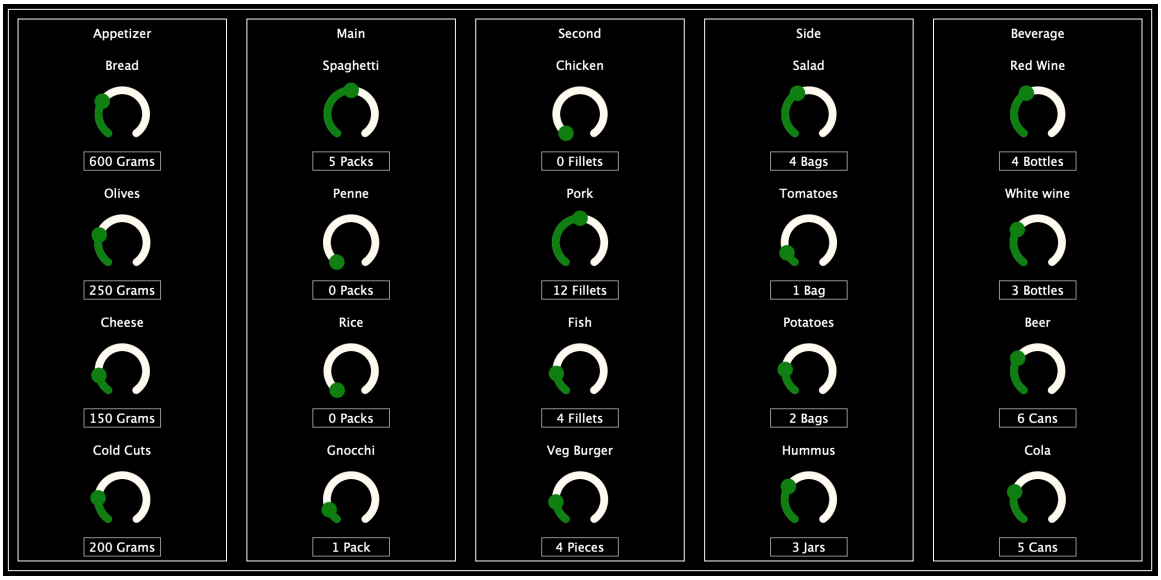

Figure 3: *GUI developed for the study. Slider values are random and for illustration only.*

Table 1: *Macro groups with associated items*

Appetizer	Main	Second	Side	Beverage
bread	spaghetti	chicken	salad	red wine
olives	penne	pork	tomatoes	white wine
cheese	rice	fish	potatoes	beer
cold cuts	gnocchi	veg burger	hummus	cola

2.2.1. *Visual modality mapping*

We define a structure as the underlying representation of the interface that organizes data and visual elements in a format suitable for screen-reader traversal and keyboard navigation. Following Kim et al. [24], we refer to information granularity as the various levels of detail through which users can explore content, from high-level summaries down to specific parameters. In practice, this means providing short labels and an overview at the top level, with deeper branches allowing details on demand. Additionally, we ensure sufficient color contrast for low-vision users, so the interface remains visually accessible as well. The final GUI layout (Figure 3) emerges from the same hierarchical structure shown in Table 1, arranged into 5 columns and 4 rows, with panels that reflect the macro groupings. Each child node is a rotary `Slider` placed vertically to its parent, arranged with JUCE's standard layout classes (`Grid` and `FlexBox`) and registered within the class framework `AccessibilityHandler`.

2.2.2. *Auditory modality mapping*

For the auditory side, each node's position is derived from the hierarchical tree structure, mapped into azimuth and elevation coordinates for Ambisonics encoding. At runtime, when VoiceOver reads a node, we intercept its text-to-speech event and traverse the hierarchy to determine both a macro-group index (azimuth offset) and a child index (elevation offset). We then send these coordinates via OSC to Reaper, where the IEM *StereoEncoder* plug-in spatializes each spoken element in a 7th order Ambisonics field, transposing the generalized user interface shown in Table 1 on a sphere.

Azimuth mapping. Although the *StereoEncoder* could pan a source 360° around the listener, we limit the horizontal plane to $[-120^\circ, 120^\circ]$, avoiding excessive head rotation and preserving a proportional alignment with the 2D interface layout. Each macro group is assigned a distinct azimuth angle from left to right: $\{120^\circ, 60^\circ, 0^\circ, -60^\circ, -120^\circ\}$.

Elevation mapping. Within each macro group, child elements (sliders) keep the parent's azimuth but vary in elevation within $[-30^\circ, 60^\circ]$, both to mirror the on-screen vertical layout and to prevent overly large head tilts. From bottom to top, four sliders occupy $\{-30^\circ, 0^\circ, 30^\circ, 60^\circ\}$, paralleling their on-screen layout.

2.3. Head-tracked binaural rendering

We pair a binaural rendering pipeline with real-time head tracking to provide a stable, external acoustic reference for each spoken interface element. We adopt a compact inertial approach for its affordability and low latency. We use an InvenSense MPU-9250 IMU, which includes a 3-axis gyroscope, 3-axis accelerometer, and 3-axis magnetometer with 16-bit resolution over I^2C, to measure yaw and pitch at rates sufficient for head rotation speeds up to 90°/s without introducing perceptible lag [25]. Although the MPU-9250 is no longer commercially produced, similar MEMS-based IMUs (e.g. InvenSense ICM-20948) provide comparable resolution and reliability for this application. We attach the IMU to the user's headphone band (AKG K52) and employ an Espressif ESP32 microcontroller to read and process sensor data. Specifically, we fuse raw gyroscope, accelerometer, and magnetometer readings into stable yaw, pitch, and roll angles [26], which are then transmitted via OSC to the IEM's *SceneRotator* plug-in. This plug-in rotates the Ambisonics field in real time according to the user's head orientation, so that each Ambisonics-encoded element remains locked to the same external directions when the listener turns their head. This ensures that if the user physically turns

their head to the left, the plug-in rotates the Ambisonics scene to the right, maintaining the illusion of a stable, externally anchored sound source.

Finally, the *BinauralDecoder* plug-in applies a generic Neumann KU100 dummy head HRTF to convert the Ambisonics mix into a stereo headphone signal, enabling head-tracked 3D audio. Prior work suggests that head tracking alone enhances spatial cues, making generic HRTFs adequate for stable daily use [27].

3. EXPERIMENTAL DESIGN

Our goal is to assess whether a head-tracked Ambisonics approach, integrated into screen readers, can reduce cognitive load, reinforce the user's mental map of the interface and improve performance when navigating a desktop app without visual access. Specifically, we aim to compare the spatialized setup against standard VoiceOver in terms of memory retention, task efficiency, and subjective user experience. To investigate this, we adopted a between-groups experimental design with two conditions:

- **Control Group (C-Group):** Standard VoiceOver.
- **Experimental Group (E-Group):** VoiceOver spatialized in Ambisonics with head-tracked binaural rendering.

A total of 32 volunteers (16 per group) were recruited among university students and staff. Aware of this approach's limitations, we selected sighted participants as a practical, rapid probe in the early exploratory phase of our design research [28]. Insights from this formative step will guide subsequent iterations that actively involve expert screen-reader users. Participants were on average 29.5 years old, and none reported prior experience with immersive audio or screen readers. Assignment to the control or experimental condition was randomized, ensuring an equal number of participants in each group. All data collection was carried out anonymously: each participant received a code (e.g., *A1* or *B2*) for identification, and no personal information was retained beyond these codes and age. The study was conducted in accordance with institutional guidelines and with the informed consent of all participants.

3.1. Experimental protocol

Building on previous work using tangible grids for interface exploration and reconstruction [29], we developed the following experimental protocol. Participants were seated comfortably in front of a standard computer keyboard, but had no visual access to the monitor, each session lasted approximately 30 minutes and consisted of 4 phases.

Interaction Briefing. Both groups received a short tutorial on how to navigate the interface via keyboard. Participants used the arrow keys (`Up`, `Down`, `Left`, `Right`) to move focus among sibling controls, while pressing the + or − keys jumped one level up or down in the hierarchy (e.g., entering or exiting a macro group, set a specific slider's value). In the E-Group, we additionally explained the head-tracker usage: rotating one's head in physical space does not move the perceived location of each control. Once participants confirmed they understood the interaction paradigms, we proceeded.

Exploration Task. Participants were asked to create a shopping list for a dinner of 12 people, choosing at least one dish from each course but otherwise free to pick items and quantities. They used keyboard input (and head-tracked audio for the E-Group) to navigate the interface, selecting or skipping items as they wished.

Interface Reconstruction. We provided an 8×8 tangible grid (50 cm×50 cm) and a set of 3D-printed circular tags (diameter 5 cm), each labeled with the name of the single UI controls. Participants were asked to place the tags on the grid according to their mental model of the application's layout. The number of tags matched the overall number of controls in the interface (20 in total), but we did not require them to place all tags, only those they recalled and felt certain about.

User Experience Questionnaires. Finally, both groups completed the same survey designed to obtain complementary information on workload, perceived usability, and personal opinions.

3.2. Quantitative data collection

We recorded four main quantitative measures to evaluate each participant's performance and efficiency:

Keyboard Interactions. We tracked the number of key presses during the exploration task (shopping list), detecting each shift in focus among the UI elements.

Exploration Time. The total time participants spent navigating the interface to complete the exploration task.

Reconstruction Time. Once participants began the interface reconstruction, we measured how long they took to arrange the 3D-printed tags on the grid until they declared themselves finished.

Reconstruction Accuracy. We compared each placed tag's row and column coordinates (r', c') on the grid with its correct position (r, c) in the UI, computing the error via the *Manhattan distance*:

$$d_{\text{Manhattan}} = |r - r'| + |c - c'| \tag{1}$$

This metric sums horizontal and vertical displacements, reflecting how many discrete *steps* one would take to reach the correct cell. If a participant did not place a certain item on the board at all, we assigned the maximum possible Manhattan distance (d = 7), effectively treating it as if placed in the worst possible cell. We then summed these distances across all items to derive each participant's final accuracy score.

3.3. User experience data collection

At the end of the experiment, participants completed two established questionnaires to gauge their subjective experience: NASA–TLX for perceived workload and the System Usability Scale (SUS) for overall usability. The NASA Task Load Index [30] assesses six dimensions: Mental Demand, Physical Demand, Temporal Demand, Performance, Effort, and Frustration. Each dimension was rated on an 11-step scale (0–10). Additionally, participants made pairwise comparisons among the six dimensions to derive custom weights reflecting which factors they felt were most critical. This weighted NASA–TLX design aims to capture not just how demanding each aspect was, but also how important participants perceived each aspect to be in the given task. In our study, the reference task was the *interface reconstruction*, so the NASA–TLX scores indicate how cognitively and physically taxing users found that process. Following the standard instruction, by summing and rescaling those item scores, we obtain a final value ranging from 0 to 100, where higher scores indicate higher perceived workload.

The System Usability Scale [31] is a 10-item questionnaire commonly used to measure a product's overall usability. Each

item uses a 5-point Likert scale ranging from *strongly disagree* to *strongly agree*, covering aspects such as ease of use, consistency, and the participant's confidence in operating the system. Also here, following the standard instruction, we obtain a final SUS value ranging from 0 to 100, where higher scores indicate better perceived usability. Finally, participants answered 2 open-ended questions:

- (Q1) *Which strategies did you adopt to remember the layout of the controls?*
- (Q2) *If you could change one aspect of the system, what would it be?*

All raw collected data (e.g. interface reconstruction photos and questionnaires) are available for reference on a public repository[1].

4. RESULTS AND DISCUSSION

Compared to the experimental group (E-Group), participants in the control group (C-Group) made on average 52% more layout errors (26.6 vs. 17.5), spent 17% less time exploring the interface (241 s vs. 291 s), but required 9% more time to reconstruct it (210 s vs. 193 s), and performed 14% fewer key presses (51 vs. 59). To assess the statistical significance of these differences, we performed a Shapiro–Wilk test for normality on each group [32], followed by either a two-sample t-test (if normally distributed) or a Mann–Whitney test (if non-normal) [33]. In every comparison, we generally found $p > 0.1$, indicating no statistically significant differences in overall workload, usability, or objective metrics such as exploration time or key presses across the two groups. Nevertheless, reconstruction accuracy analysis reveals an interesting trend: participants in the head-tracked group, despite reporting similar NASA–TLX and SUS scores, achieved more precise layouts of the interface elements. The following subsections present our quantitative measurements and user experience data (including open-ended responses), illustrating how spatial audio cues may have influenced recall strategies and led to fewer layout misplacements.

4.1. Quantitative results

Tables 2 and 3 summarize the quantitative metrics we gathered from both the control (C-Group) and experimental (E-Group) participants.

Table 2: Mean values and standard deviations for each group

	Nr. Click	Explr. Time	Reconstr. Time	Manhattan Error
C-Group	51±15	241±86 s	210±123 s	26.6±22.8
E-Group	59±16	291±108 s	193±69 s	17.5±12.7

Although the E-Group shows slightly higher average exploration time and key presses, no statistically significant differences emerged ($0.15 < p < 0.18$ for all comparisons). By contrast, the Manhattan Error reveals a more notable gap (26.6 vs. 17.5), indicating that the head-tracked group tended to reconstruct the interface more accurately overall.

[1] Available at: https://github.com/GiuseppeBergamino/DAFx25

Table 3: Total number of reconstruction features

	Missing elements	Spatialized reconstructions	Rotated reconstructions
C-Group	24	5	6
E-Group	6	8	2

Table 3 presents the total number of features identified, further illustrating behavioral differences in how the two groups approached the interface layout. The E-Group omitted fewer elements in total (6 vs. 24 items) and produced more *spatialized reconstructions*, meaning they distributed controls across the full 8×8 grid, often mirroring the head-tracked binaural spread. Interestingly, even though the C-Group lacked auditory cues for spatial placement, some participants nonetheless spaced controls in a similarly expanded layout, whereas others compacted them or inadvertently rotated the grid by 90° (6 vs. 2 cases). Although these frequencies did not achieve formal statistical significance, they point to more robust spatial recall strategies within the E-Group, aligning with the observations from open-ended feedback (Section 4.2).

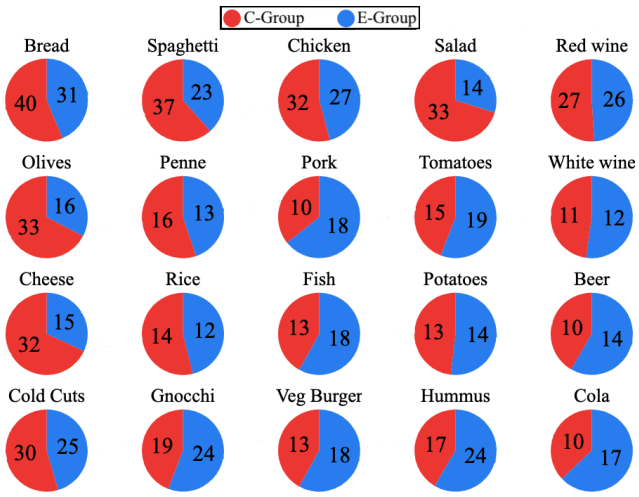

Figure 4: *Per-item reconstruction errors. Each pie chart shows the cumulative positional errors for a specific UI control, calculated across all 16 participants per group. The layout replicates the original interface structure.*

Figure 4 provides a more granular view of how much each individual control was misplaced, following the Manhattan distance metrics. Each pie chart corresponds to a specific control in the interface and is placed in the figure according to its original location in the layout (Figure 3). The red and blue segments respectively represent the cumulative errors made by the C-Group and the E-Group for that particular control, aggregated over all 16 participants. These values reflect the errors as *seen* from the perspective of each item, summing all red (or blue) values and dividing by 16 yields the average Manhattan error per group reported in Table 2.

Notably, the highest error counts often appear in the *first* item of each column, implying that participants occasionally skipped VoiceOver's verbose introductory announcements (e.g., *In group, four elements...*) upon entering a new category. In many such cases, the first item was completely omitted from the reconstruction, resulting in an assigned distance error of 7 (Equation 1).

This behavior was more pronounced in C-Group, which probably lacked the auditory spatial cues and appeared more inclined to fast-forward through the interface. In contrast, the E-Group, while still making mistakes, was less likely to miss these top items, but tended to err on the *bottom* items in each column, where our mapping placed controls at -30° elevation. This suggests that the chosen overly steep negative angle may have made those bottom controls harder to localize, leading to a different pattern of errors. Collectively, these pie charts reinforce our findings: the E-Group committed fewer overall errors (Table 2), and differences in *where* errors occurred point toward the value of spatial audio cues in guiding navigational strategies.

It is important to note that these findings reflect only the performance of sighted participants. Previous studies suggest that blind people often surpass sighted users in auditory spatial localization [34], displaying heightened sensitivity to binaural cues [35]. Based on these works, we hypothesize that future evaluations involving blind and low-vision users may reveal even more accurate interface reconstructions, with further reductions in spatial positioning errors.

4.2. User experience results

Participants' subjective workload is assessed using NASA task load index (TLX) and overall usability using the System Usability Scale (SUS). NASA–TLX scores range from 0 (minimal workload) to 100 (extremely high workload), whereas SUS ranges from 0 (poor usability) to 100 (excellent usability). Results indicate that both groups reported comparable values on both scales.

For the NASA–TLX, the control group (C-Group) averaged 39.75 (±17.24), while the experimental group (E-Group) averaged 38.11 (±17.09), suggesting that the spatialized audio interface did not impose additional or reduced cognitive workload. On the SUS, the C-Group reported an average score of 79.69 (±11.18) compared to 78.91 (±16.96) in the E-Group, with both scores indicating a high level of perceived usability. Statistical comparisons revealed no significant differences between groups (all $p > 0.1$), confirming comparable subjective experiences.

Although the two groups showed similar qualitative ratings, the open-ended questions (Section 3.3) offered further insights. For Q1, 50% (8/16) of E-Group participants explicitly mentioned the use of different kind of memorization strategies. Conversely, the majority of the C-Group 69% (11/16) provided non-specific or generic answers, suggesting less structured recall actions. In Q2, the E-Group participants predominantly suggested targeted improvements to navigation modalities 38% (6/16) or enhancements to auditory feedback quality 19% (3/16). By contrast, 75% (12/16) of C-Group participants gave broad or vague suggestions, and 25% (4/16) focused on modifying the speech speed and rhythm, with fewer direct ideas on structural changes.

These findings suggest that despite similar workload and usability scores, the E-Group exhibited more structured engagement with the system, leading to more precise and constructive feedback for system improvements.

5. CONCLUSIONS

In this study, we developed a prototype that spatializes screen-reader output in a spherical audio field, augmented by a head tracker for real-time orientation feedback. Our implementation uses accessibility handler by JUCE for GUI labeling, macOS VoiceOver for text-to-speech, and an Ambisonics pipeline (IEM plug-ins in Reaper) to deliver head-tracked binaural audio. Although we found no statistically significant differences in perceived workload or usability, participants in the head-tracked group consistently produced more accurate reconstructions, suggesting that immersive audio cues can foster structured memorization strategies. However, sample size (16 per group) and the involvement of sighted participants likely restricted our capacity to detect finer effects.

Future plans involve integrating these components more deeply into non-visual workflow environments (e.g., music production user interfaces) and extending compatibility to other screen readers and operating systems. We also intend to co-design and test the system with blind and low-vision users, bringing them in not only for evaluation, but as active partners in every subsequent development cycle. Guided by a design-based research approach, their feedback will shape iterative refinements (e.g., correcting error-prone auditory mappings) so that head-tracked binaural feedback becomes a robust, practical enhancement for accessible user interfaces.

6. ACKNOWLEDGMENTS

Giuseppe Bergamino's PhD scholarship is funded by the Marche Region within the framework of the "Innovative PhD Programmes", under the Regional Programme of the European Social Fund Plus (PR FSE+) 2021–2027.

7. REFERENCES

[1] World Health Organization, "Blindness and vision impairment," Available at https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment, accessed June 9, 2025.

[2] S. Oviatt, *The human-computer interaction handbook*, chapter Multimodal Interfaces, pp. 439–458, CRC Press, Boca Raton, USA, 2007.

[3] A. D N Edwards, "The design of auditory interfaces for visually disabled users," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, Washington D.C., USA, May 15-19, 1988, pp. 83–88.

[4] W. W Gaver, "The sonicfinder: An interface that uses auditory icons," *Human–Computer Interaction*, vol. 4, no. 1, pp. 67–94, Jan. 1989.

[5] M. M Blattner, D. A Sumikawa, and R. M Greenberg, "Earcons and icons: Their structure and common design principles," *Human–Computer Interaction*, vol. 4, no. 1, pp. 11–44, Jan. 1989.

[6] S. Brewster, "The design of sonically-enhanced widgets," *Interacting with Computers*, vol. 11, no. 2, pp. 211–235, Feb. 1998.

[7] C. Frauenberger and M. Noisternig, "3d audio interfaces for the blind," in *International Conference on Auditory Display*, Boston, USA, Jul. 6-9, 2003, pp. 280–283.

[8] F. Ribeiro, D. Florencio, P. Chou, and Z. Zhang, "Auditory augmented reality: Object sonification for the visually impaired," in *2012 IEEE 14th international workshop on multimedia signal processing (MMSP)*, Banff, Canada, Sept. 17-19, 2012, pp. 319–324.

[9] K. Crispien, W. Würz, and G. Weber, "Using spatial audio for the enhanced presentation of synthesised speech within screen-readers for blind computer users," in *International Conference on Computers for Handicapped Persons*, Vienna, Austria, Sept. 17-19, 1994, pp. 144–153.

[10] S. Goose and C. Möller, "A 3d audio only interactive web browser: using spatialization to convey hypermedia document structure," in *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, Orlando, USA, Oct. 30-Nov. 5, 1999, pp. 363–371.

[11] J. Sodnik, Jakus G, and S. Tomažic, "Enhanced synthesized text reader for visually impaired users," in *2010 Third International Conference on Advances in Computer-Human Interactions*, Saint Maarten, Netherlands, Feb. 10-15, 2010, pp. 91–94.

[12] M. Morris, J. Johnson, C. Bennett, and E. Cutrell, "Rich representations of visual content for screen reader users," in *Proceedings of the 2018 CHI conference on human factors in computing systems*, Montréal, Canada, Apr. 21-26, 2018, pp. 1–11.

[13] J. Zong, C. Lee, A. Lundgard, J. Alan, J. Jang, D. Hajas, and A. Satyanarayan, "Rich screen reader experiences for accessible data visualization," in *Eurographics Conference on Visualization (EuroVis) 2022*, Rome, Italy, Jun. 13-17, 2022, pp. 15–27.

[14] A. Chheda-Kothary, D. Rios, K. S Smith, A. Reyna, C. Zhang, and B. A Smith, "Understanding blind and low vision users' attitudes towards spatial interactions in desktop screen readers," in *Proceedings of the 25th International ACM SIGACCESS Conference on Computers and Accessibility*, New York, USA, Oct. 22-25, 2023, pp. 1–5.

[15] J. Lazar, A. Allen, J. Kleinman, and C. Malarkey, "What frustrates screen reader users on the web: A study of 100 blind users," *International Journal of human-computer interaction*, vol. 22, no. 3, pp. 247–269, Mar. 2007.

[16] WebAIM, "Screen Reader User Survey nr.10 Results," Available at https://webaim.org/projects/screenreadersurvey10/, accessed June 9, 2025.

[17] C. Frauenberger, V. Putz, R. Holdrich, and T. Stockman, "Interaction patterns for auditory user interfaces," in *Proceedings of ICAD 05-Eleventh Meeting of the International Conference on Auditory Display*, Limerick, Ireland, Jul. 6-9, 2005, pp. 154–160.

[18] W. C. Payne, A. Y. Xu, F. Ahmed, L. Ye, and A. Hurst, "How blind and visually impaired composers, producers, and songwriters leverage and adapt music technology," in *Proceedings of the 22nd International ACM SIGACCESS Conference on Computers and Accessibility*, Virtual Event, Greece, Oct. 26-28, 2020, pp. 1–12.

[19] M. Steadman, C. Kim, J. Lestang, D. Goodman, and L. Picinali, "Short-term effects of sound localization training in virtual reality," *Nature Scientific Reports*, vol. 9, no. 1, pp. 18284, Dec. 2019.

[20] C. Valzolgher, C. Campus, G. Rabini, M. Gori, and F. Pavani, "Updating spatial hearing abilities through multisensory and motor cues," *Cognition*, vol. 204, no. 1, pp. 104409, Nov. 2020.

[21] Apple, "VoiceOver Guide," Available at https://support.apple.com/guide/voiceover/welcome/mac, accessed June 9, 2025.

[22] IEM, "IEM Plug-in Suite," Available at https://plugins.iem.at/, accessed June 9, 2025.

[23] JUCE, "JUCE Accessibility Handler," Available at https://docs.juce.com/master/classAccessibilityHandler.html, accessed June 9, 2025.

[24] N. W. Kim, S. C. Joyner, A. Riegelhuth, and Y. Kim, "Accessible visualization: Design space, opportunities, and challenges," in *Eurographics Conference on Visualization (EuroVis) 2021*, Zurich, Switzerland, Jun. 14-18, 2021, pp. 173–188.

[25] P. Franček, K. Jambrošić, M. Horvat, and V. Planinec, "The performance of inertial measurement unit sensors on various hardware platforms for binaural head-tracking applications," *Sensors*, vol. 23, no. 2, pp. 872, Jan. 2023.

[26] Hideakitai, "Github library repository," Available at https://github.com/hideakitai/MPU9250, accessed June 9, 2025.

[27] O. S Rummukainen, T. Robotham, and E. AP Habets, "Head-related transfer functions for dynamic listeners in virtual reality," *Applied Sciences*, vol. 11, no. 14, pp. 6646, July 2021.

[28] G. W Tigwell, "Nuanced perspectives toward disability simulations from digital designers, blind, low vision, and color blind people," in *Proceedings of the 2021 CHI conference on human factors in computing systems*, Virtual event, Japan, May. 8-13, 2021, pp. 1–15.

[29] J. Li, Z. Yan, E. H. Jarjue, A. Shetty, and H. Peng, "Tangiblegrid: Tangible web layout design for blind users," in *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, Bend, USA, Oct. 29-Nov. 2, 2022, pp. 1–12.

[30] S. G. Hart and L. E. Staveland, "Development of nasa-tlx (task load index): Results of empirical and theoretical research," in *Advances in psychology*, vol. 52, pp. 139–183. Elsevier, 1988.

[31] J. Brooke, "Sus-a quick and dirty usability scale," *Usability evaluation in industry*, vol. 189, no. 194, pp. 4–7, Aug. 1996.

[32] S. S Shapiro and M. B Wilk, "An analysis of variance test for normality (complete samples)," *Biometrika*, vol. 52, no. 3, pp. 591–611, Mar. 1965.

[33] T. W MacFarland and J. M Yates, "Mann–whitney u test," *Introduction to nonparametric statistics for the biological sciences using R*, vol. 6, no. 4, pp. 103–132, July 2016.

[34] N. Lessard, M. Paré, F. Lepore, and M. Lassonde, "Early-blind human subjects localize sound sources better than sighted subjects," *Nature*, vol. 395, no. 6699, pp. 278–280, Sept. 1998.

[35] M. E Nilsson and B. N Schenkman, "Blind people are more sensitive than sighted people to binaural sound-location cues, particularly inter-aural level differences," *Hearing research*, vol. 332, no. 2, pp. 223–232, Feb. 2016.