



A11yBoard: Making Digital Artboards Accessible to Blind and Low-Vision Users

Zhuohao (Jerry) Zhang

The Information School | DUB Group
University of Washington, Seattle, WA, USA
zhuohao@uw.edu

Jacob O. Wobbrock

The Information School | DUB Group
University of Washington, Seattle, WA, USA
wobbrock@uw.edu

ABSTRACT

Digital artboards, which hold objects rather than pixels (e.g., Microsoft PowerPoint and Google Slides), remain largely inaccessible for blind and low-vision (BLV) users. Building on prior findings about the experiences of BLV users with digital artboards, we present a novel tool called *A11yBoard*, an interactive multimodal system that makes interpreting and authoring digital artboards accessible. A11yBoard combines a web-based drawing canvas paired with a mobile touch screen device such as a tablet. The mobile device displays the same canvas and enables risk-free spatial exploration of the artboard via touch and gesture. Speech recognition, non-speech audio, and keyboard-based commands are also used for input and output. Through a series of pilot studies and formal task-based user studies with BLV participants, we show that A11yBoard provides (1) intuitive spatial reasoning about two-dimensional objects, (2) multimodal access to objects' properties and relationships, and (3) eyes-free creating and editing of objects to establish their desired properties and positions.

CCS CONCEPTS

• **Human-centered computing** → **Human computer interaction (HCI)**; **Accessibility technologies**.

ACM Reference Format:

Zhuohao (Jerry) Zhang and Jacob O. Wobbrock. 2023. A11yBoard: Making Digital Artboards Accessible to Blind and Low-Vision Users. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*, April 23–28, 2023, Hamburg, Germany. ACM, New York, NY, USA, 17 pages. <https://doi.org/10.1145/3544548.3580655>

1 INTRODUCTION

Most popular design processes, such as design thinking [10], explicitly prioritize users across the phases of user research, ideation, and evaluation. When it comes to blind and low-vision (BLV) users, prior research has shown ways to include them in these three phases of the design process [1, 14, 59]. But when it comes to the *creation* phase of the design process, where content, products, or services are prototyped or developed, less has been done to include BLV users [42]. Indeed, some prior work has explored enabling BLV users to create digital content like sonifications [58], programming

code [17], and web design [48], but there has not yet been a significant research making visual content authoring accessible, such as for making presentations, developing wireframes, laying out posters, or sketching storyboards. The result is that people with disabilities may be *consumers* of content, but they are not yet widely empowered to become *creators* of content.

One type of prevalent content creation tool in educational and professional settings is a tool for slide design, such as Microsoft PowerPoint, Google Slides, or Apple Keynote [60]. These tools usually consist of a digital canvas (sometimes known as a digital “artboard,” the term we use here) and contain objects like text boxes, shapes, arrows, images, tables, and so on. (By contrast, digital “paint canvases” accept pixels, like in Adobe Photoshop or Microsoft Paint.) For BLV users, given the highly visual nature of artboards, *interpreting* existing artboards and *generating* new ones remain largely inaccessible. Although prior research has investigated how to support accessible drawing [8, 23, 33, 40], working with artboard content still poses numerous accessibility issues for BLV users [60]. And as the prevalence of digital artboards in professional and educational settings is high, making them accessible is of utmost importance.

A11yBoard seeks to remedy the problems encountered by BLV users of digital artboards. Prior work [60] has investigated these accessibility challenges thoroughly—major issues include high cognitive load; difficulty determining object relationships, positions, and properties; and the uncertainty about the success (or failure) of operations and commands, such as changing an object's color or its position relative to another object. It is extremely difficult for anyone to build and maintain an accurate mental model of an artboard and all its contents, and to understand complex object relationships like object overlap (“Z-order”), objects' relative positions, orientations, sizes, and so on. A fundamental challenge is that conventional screen readers were developed primarily to handle text, which is a linear one-dimensional (1-D) information stream. But a digital artboard is inherently a two-dimensional (2-D) space containing potentially many 2-D objects, making it ill-suited to today's screen readers. For example, Microsoft Windows's built-in screen reader, *Windows Narrator*, when faced with a Microsoft PowerPoint slide, reads out all artboard objects based only on their Z-order, regardless of where those objects are located or whether they are visually overlapping. Presumably, the reason for this choice was that the Z-ordering of objects is a well-defined 1-D information stream, even if it is mostly useless in a 2-D information space. This example is just one of the many challenges modern screen readers face when encountering 2-D spaces like digital artboards.

To address the aforementioned challenges, we present *A11yBoard*, an interactive multimodal multi-device system that provides for accessible interpreting and authoring of digital artboards for BLV users (see Figure 1). A11yBoard combines a web-based application,



This work is licensed under a Creative Commons Attribution International 4.0 License.

CHI '23, April 23–28, 2023, Hamburg, Germany
© 2023 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9421-5/23/04.
<https://doi.org/10.1145/3544548.3580655>

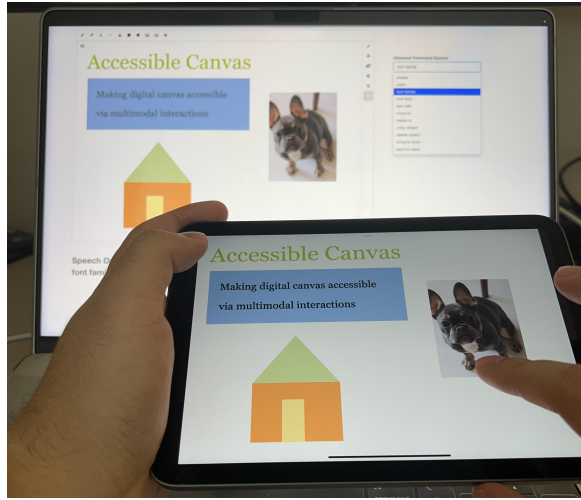


Figure 1: A11yBoard is a multimodal multi-device system consisting of a web-based artboard tool, usually appearing on a desktop or laptop, mirrored onto a touch display. A11yBoard supports interpreting an existing artboard via risk-free touch and gesture exploration on the touch display, from which it outputs information about objects using audio tones and speech. For example, when a user’s finger crosses into an object, a “step up” tone is played. When a user split-taps on an object [34], speech output describes the object. When a user’s finger crosses out of an object, a “step down” tone is played. A11yBoard also supports generating and modifying an artboard’s contents using speech recognition and desktop keyboard commands. Intentionally, the mouse is never used.

usually shown on a desktop or laptop computer, with a mobile touch screen device like a smartphone or tablet. The web application supports common artboard authoring features like creating and editing text boxes, images, and shapes, and the artboard is mirrored from the user’s desktop onto their touch screen, enabling risk-free spatial exploration of the artboard via touch and gesture [34, 35, 38]. Furthermore, audio tones and custom screen reader outputs are played in response to a user’s touches and gestures. In addition, speech recognition and voice assistant interactions are also used for accessing objects’ information and editing object properties. Finally, the desktop keyboard is used with a custom search-driven command line to access various commands and properties when these properties are not easily manipulable using touch, gesture, and voice. And intentionally, the mouse plays no role in A11yBoard. The result is a holistic multi-device multimodal user experience that enables BLV users to consume and create digital artboards.

To evaluate A11yBoard, we conducted iterative preliminary study sessions as well as a formal user study. In our multiple preliminary study sessions with three blind participants, we received positive feedback on the usefulness and naturalness of A11yBoard’s multimodal interactions. We also received constructive criticisms that drove our iterations of A11yBoard’s design, as described in Section 5. Subsequently, we conducted a formal user study with

seven blind participants and one low-vision participant (see Table 1), most of whom had some prior experience in accessing or authoring artboards (in this case, slide presentations). Using a structured series of usability tasks from prior work [60], we showed that participants completed artboard interpreting and authoring far better than has been reported without A11yBoard. Post-study interviews indicated that A11yBoard provides: (1) intuitive spatial reasoning about two-dimensional objects; (2) multimodal access to objects’ properties and relationships; and (3) eyes-free creating and editing of objects to establish their desired properties and positions. These features combine into an accessible and effective interaction for both *interpreting* and *authoring* digital 2-D artboards.

Multiple participants reported that A11yBoard should be further incorporated into existing tools like Microsoft PowerPoint or Google Slides. Participants also conveyed how A11yBoard’s interactions could be applied beyond artboards. For example, the ideas embodied in A11yBoard have the potential to generalize to other 2-D spaces like web pages or mobile user interfaces by making such spaces “touchable” and “speakable” via multimodal multi-device interactions.

2 RELATED WORK

Prior work related to *A11yBoard* resides mainly in two categories, which we review below. The first is using multiple modalities to enable non-visual interactions for BLV users. The second is research into making creativity tools more accessible to BLV users.

2.1 Multimodal Interactions for Blind or Low-Vision Users

Different input and output modalities have been introduced to enable non-visual interactions for BLV users. Here, we organize non-visual interactions for BLV users specifically into three categories: audio, tactile, and multimodal interactions. Unsurprisingly, the realm of non-visual interactions consists of extensive work such that a complete review is prohibited. Therefore, we focus on prior research concerned with 2-D spaces, like screens, maps, diagrams, images, documents, and so on.

First, audio interactions, including speech recognition, text-to-speech, and implicit audio, have been widely adopted in assistive technologies for BLV users. Voice assistants (e.g., Apple Siri, Amazon Alexa) and screen readers (e.g., VoiceOver [2], NVDA [52], JAWS [30]) have enabled BLV users to access visual elements via speech output. Besides these commercially available tools, prior research has also investigated other auditory techniques to make virtual 2-D spaces more accessible, like user interface design [29], graphs [9, 12, 13, 62, 63], maps [15, 16, 64] and documents [46]. Kane et al. [34] created Slide Rule, the first finger-driven touch-based screen reader, which introduced what later became known as the split-tap gesture [45]. Kane et al. [35] employed similar techniques for large tabletop displays in Access Overlays. Meanwhile, EdgeSonic [74] proposed automatic sonification of graphical information from apps and images using a “distance-to-edge” map to guide users towards detected edges. Similarly, RegionSpeak [75] also provided spatial descriptions of complex image regions. Blenkhorn et al. [7] and Kennel et al. [39] also created diagram readers to make schematic diagrams accessible to BLV users. Lee

et al. [47] developed ImageExplorer, which employed audio feedback to let BLV users explore the spatial layout and information hierarchies of images.

Second, tactile and haptic interfaces have also been investigated to support non-visual interactions. Unlike traditional keyboard and mouse interactions, tactile and haptic interactions allow for more intuitive representations of graphical and operational information [3, 4, 24, 36, 44, 49, 50, 53, 73]. Prior research has also studied various forms of tactile and haptic feedback for BLV users to interact with maps [65, 66] and graphs [40, 69]. The Tactile Graphics project [31, 43] sought to make textbooks more accessible through creating embossed scientific plots. However, these customized solutions often require specialized hardware and cannot be easily used at scale. In contrast, Ducasse et al. [16] introduced hybrid physical and digital interactive maps used to make geographical data accessible to BLV users. They also mentioned advantages of digital tablets [11], which do not require additional (and potentially expensive) devices to make visual elements tactile. A11yBoard also adopted a similar motivation in using only off-the-shelf computers and custom software, avoiding custom or arcane hardware so as to be easily deployable.

Third, audio and tactile interactions are often employed together in multimodal designs to create more accessible experiences for BLV users [56, 68, 69]. For example, McSig [56] combined haptic and audio output to support visually-impaired students to learn character shapes and handwriting by copying teacher’s pen input. Wall and Brewster [69] developed a system that uses audio feedback and haptics to support a point-and-click type interaction for BLV users. Other combinations of modalities also have been used as assistive technologies for everyday tasks, such as using touch, audio, gesture, and vibrations [19, 32]. Similarly, A11yBoard also uses multimodal interactions to support accessing 2-D digital artboards, an approach not yet investigated.

2.2 Accessible Creativity Tools

As noted in our introduction, most modern design processes prioritize users in research, ideation, and evaluation [10, 22, 51]. However, there has not yet been a significant push to include users in the *creation* phase of the design process [42]. Rightly so, much focus has been on making content consumable by people with disabilities [47, 54, 55, 63, 70]. But people with disabilities deserve more than just to be *consumers* of design outputs; they deserve to be *creators* of such outputs as well. But the lack of accessible creativity tools is a glaring reminder of the need to empower people with disabilities in the prototyping and development phases of the design process.

Indeed, there has been much prior work that aims to enable BLV users to consume accessible 2-D media, and without doubt this work is vital. For example, VoxLens [63] enables BLV users to obtain holistic summaries of online data visualizations, playsonified versions of that data, and interact with visualizations in a “drilled-down” manner using voice-activated commands. Wang et al. [70] used machine learning models in SciA11y, which extracts the content of scientific Adobe PDF documents and renders this content as accessible HTML, with added navigational features to support screen readers. Other research has also made images [47], presentation slides [55], and even videos [54] more accessible to

BLV users. Certainly, making content consumable is a crucial step towards making artboards accessible, but it requires a much further step to make the authoring of artboards accessible as well.

Prior research has also investigated how assistive technologies can support digital drawing. As an early effort, TDraw [41] used a heat pen, swell paper, and continuous tactile feedback to support blind users. Verbal information via speech input was also used to augment the drawings, which were then automatically rendered digitally after being performed on paper. However, TDraw focused on pixel-based drawing rather than object- or vector-based drawing. As for drawing objects or vectors, AudioDraw [23] integrated touch screen input and text-to-speech feedback to allow users to select and place pre-made shapes. Kamel and Landay [33] introduced a novel input method for BLV users to draw lines, polygons, and circles via keyboard commands using a 3×3 grid. EyeDraw [28] relied upon eye-tracking to enable drawing by people with motor impairments. Inspired by EyeDraw, VoiceDraw [25] built atop the Vocal Joystick [6] to enable fluid, continuous paintbrush control through the continuous articulation of vowel sounds. Beyond drawing, Potluri et al. [57] designed a multimodal system for BLV users to modify user interface designs and code using a touch screen.

The above work points toward a burgeoning need to make creativity tools accessible by designing new interaction techniques and incorporating them into functional systems. A11yBoard does just this, providing a holistic system for addressing both artboard interpretation and artboard generation. It combines multiple devices as well as multiple modalities, both input and output, to map the right kinds of actions to the right kinds of inputs. We now turn towards a more in-depth discussion of A11yBoard, its objectives, and its capabilities.

3 DESIGN CONSIDERATIONS

Drawing on prior work [60], we formulated six design considerations that guided our design of *A11yBoard*. The first three considerations are directly from prior work’s [60] main findings; the second three were formulated through initial discussions with our participants, and from participants’ comments in prior work.

1. Reduce cognitive load: For 2-D artboards full of 2-D content, using a screen reader with a 1-D reading order is ineffective and arbitrary, and results in users having to construct a mental map of object relationships, overwhelming users’ working memory. Although it is true that text boxes themselves contain 1-D content, text boxes are also 2-D objects that can reside along with other 2-D objects anywhere on the artboard. A11yBoard must reduce the cognitive load involved in interpreting an artboard by making it easier (or unnecessary) for users to remember canvas objects and their states, properties, and relationships to other objects.

2. Remove uncertainty of operation success: Sighted users typically receive visual confirmation of operation success (or failure). For example, they see an object moved, resized, rotated, or changed in color. A11yBoard must provide its users with similarly quick confirmation of operations. A corollary is to also receive quick confirmation of *unwanted* operations, and be able to undo them with ease. Ideally, A11yBoard must minimize such unwanted occurrences through “risk-free exploration,” as BLV users have expressed concerns over “messing things up” when exploring artboards.

3. Provide ways to learn object relationships: Organizing objects is an essential step in artboard authoring. For example, several text boxes, images, or shapes are often grouped together to be moved or resized simultaneously. Users often find it challenging to understand relative positions and overlapping of objects. A11yBoard must give users ways to learn relationships among objects so that they can organize them accordingly.

4. Provide for spatial reasoning: A 2-D artboard is inherently spatial and non-linear. Therefore, A11yBoard users must be able to navigate this space in two dimensions and reason about its contents spatially, not be forced into linear information streams based on conventional 1-D text-based screen readers.

5. Reveal object overlap, including occlusion: As part of supporting spatial reasoning, A11yBoard must make it possible for users to understand when objects overlap, particularly with total occlusion. As with our example of *Windows Narrator* in the introduction, screen readers might simply read out the Z-order of all objects on an artboard, but this does not reveal whether those objects actually overlap or occlude.

6. Don't ignore the visual channel: Research shows that blind users often collaborate with sighted or low-vision users [5, 37, 60] and therefore might work together using A11yBoard. Furthermore, by including visual feedback, our system can be learned by sighted users who can teach others. Low-vision users might also benefit from visual feedback together with other feedback modalities.

4 THE A11YBOARD SYSTEM

We present the design and implementation of *A11yBoard*, a multi-modal multi-device system comprising a web-based artboard, usually shown on a desktop or laptop computer, coupled with a touch screen device that mirrors the artboard and enables exploration via touch and gesture. We created A11yBoard using an iterative and participatory user-centered design process, building on findings and principles from prior work [60]. Specifically, we aimed to satisfy the six design considerations in Section 3 so as to address known barriers to access. In this section, we describe our resulting A11yBoard system. In the sections that follow, we empirically evaluate A11yBoard with blind and low-vision (BLV) users.

4.1 System Overview

The core components of A11yBoard are a web-based artboard interface, usually shown on a desktop or laptop computer having a physical keyboard, and a connected touch screen device like a smartphone or tablet. A11yBoard's web-based artboard is built upon an open-sourced drawing tool named *DrawerJS* [61], which we heavily modified to create the "desktop side" of A11yBoard. DrawerJS is a JavaScript WYSIWYG HTML canvas drawing tool consisting of customary object-based drawing functions including inserting shapes, lines, text boxes, and images. Objects have properties like position, size, orientation, and color, which can be inspected and modified. Drawing functions are similar to existing artboard tools like Microsoft PowerPoint or Google Slides. Although we considered creating A11yBoard initially as an extension to Microsoft PowerPoint or Google Slides, we were uncertain whether our design decisions would ultimately require us to have more control

over these environments than application extensions typically allow; by contrast, with DrawerJS, we had full control via its entire source code.

A11yBoard employs a variety of multimodal inputs and outputs across three pieces of connected hardware: a desktop display, a touch screen device, and a desktop keyboard—note that the mouse is intentionally avoided, as it introduces numerous accessibility problems [18, 67]. The desktop display shows the artboard. The touch screen device mirrors that artboard and allows for risk-free finger-driven spatial exploration of the artboard's contents. It also provides audio output, for example, when the finger crosses into and out of objects, and speech input, for example, when asking for the nearest object to a location. The desktop keyboard can be used with what we call the "Universal Command Line," where the user can simply type a few characters related to their need and a search procedure will provide an enumerated *N*-best list of most likely commands and property values (e.g., "mag" results in the option to set an object to the color "magenta"). Whereas the touch screen allows users to spatially explore an artboard's contents through continuous touch (i.e., finger movement), the keyboard allows for execution of discrete commands, such as creating new objects (e.g., "rec" for creating a new "rectangle") or changing properties (e.g., "fro" for bringing an object to the "front" of the Z-order). We elaborate on A11yBoard's important features below.

We did not employ any commercial screen-reading software in conjunction with A11yBoard, but instead relied only upon our own speech input and audio output. A11yBoard also used Google Cloud Speech Synthesis for text-to-speech output messages [21]. For speech input, the recognition engine used by A11yBoard was provided by Google Cloud speech-to-text services [20] (see Section 4.2.2). A11yBoard was developed with a unique set of custom interactions precisely because we did not want to limit A11yBoard to what screen readers already can do. For example, Apple's VoiceOver consumes touch and gesture inputs for its own purposes and would not be reusable in A11yBoard. We discuss prospects for integrating A11yBoard with existing screen reader software in Section 8.

4.2 Supported Interactions

In this subsection, we discuss each major feature and its corresponding interaction design in detail, organized by different modalities, including touch, gesture, audio, speech, keyboard, and search.

4.2.1 Interpreting Artboards via Touch and Gesture. A core part of A11yBoard is a mobile app running on a touch screen device that enables risk-free exploration of an artboard's contents, the same contents that appear on the desktop side of A11yBoard. Our touch screen app, developed in Swift, mirrors the desktop's canvas, and supports finger-driven screen reading similar to what was pioneered by Slide Rule [34]. By mirroring the web-based artboard onto a touch screen device, A11yBoard's canvas can be explored *spatially* and *safely* through touch in an absolute-mapped 2-D fashion. Using finger-driven screen reading gives the user direct control over their spatial exploration; it also reveals objects' relative positions and sizes. Below, we discuss A11yBoard's supported touch and gesture interactions (Figure 2). Collectively, these interactions reduce cognitive load by lessening demands on users' memory and enabling

rapid, risk-free exploration of artboard objects in 2-D space (see considerations 1 and 4 in Section 3).

Single-finger exploration of artboard contents: A11yBoard supports a single finger “reading” the artboard’s contents. As feedback, A11yBoard plays two different audio tones: When a user’s finger crosses *into* an object, A11yBoard plays a “step-up” sound; similarly, when a user’s finger crosses *out of* an object, A11yBoard plays a “step-down” sound. (The specific notes for the step-up sound are F-B, and the reverse for the step-down sound.)

For more complex scenarios where objects overlap each other, A11yBoard plays these tones in quick succession. For example, A11yBoard will play two consecutive “step-up” tones when the “reading finger” moves into a text box on top of a rectangle. Similarly, when the finger moves out of this text box but is still inside the rectangle, A11yBoard will first play a “step-down” tone, and then play a “step-up” tone, indicating that the finger left the text box, but still resides in another object. This is to rapidly give users more context of objects under their reading finger when they move into and out of objects’ boundaries.

A11yBoard also responds to various finger-movement speeds differently. If the finger moves slowly enough so as to remain within an object for a moment, A11yBoard reports the basic information about the object right after audio sounds, including the object’s color name and type (e.g., “green text”). By contrast, if the finger moves quickly through objects, this feedback is cut short, and A11yBoard only plays the audio to indicate stepping into and out of objects. Relatedly, if an object’s information is being announced but the finger lifts from the screen, the speech output is abruptly cut, giving users full control through their “reading finger.” In general, A11yBoard provides audio or speech confirmation for all interactions to reduce uncertainty (see consideration 2 in Section 3).

Split-tap for more object information: While exploring the artboard canvas using a “reading finger,” users might want to access detailed information about objects, like positions, sizes, or other properties. Users might also want to select objects for further interactions, like moving or resizing. To this end, A11yBoard supports split-tap (i.e., a “second-finger tap” issued anywhere on the screen while the first finger—the reading finger—remains on the intended target [34]). When an object is split-tapped, it is selected and its size and position, in canvas-relative percentages, are announced. For example, “50% of canvas width” indicates that an object is half the width of the canvas. Similarly, “25% from canvas left” indicates that an object’s left edge is placed a quarter of the way in from the canvas’ left edge. (We found from our preliminary testing that by using canvas percentages for sizes and positions, users gained a quicker and more intuitive understanding of object arrangements than by using exact pixel dimensions.)

Split-tap object selection is required before any further interactions with that object (e.g., to move or resize it, change its color, etc.). We implemented this design to enable risk-free exploration of canvas objects, since their states cannot be modified unless a split-tap selection is performed followed by a command, as described below. To support potential collaboration with sighted or low-vision users, A11yBoard also visually highlights selected objects when they are split-tapped (see consideration 6 in Section 3).

Finally, a split-tap can also be performed on an empty canvas position where there is no object, which results in a “numb” audio

sound being played. Such a split-tap enables certain actions, e.g., creating a new object at that position.

Single-finger dwell to initiate speech recognition: Once an object is selected, A11yBoard supports further interactions including a variety of speech commands, which are actively listened for when a finger dwells anywhere on the screen. Dwelling is an “active mode” like holding down the Shift key on a keyboard or holding a button on a walkie-talkie. A11yBoard’s speech recognition remains actively listening for as long as the user’s finger dwells on the screen. A variety of speech commands are available (see Section 4.2.2, below).

Two-finger flick to discover nearby objects: To address consideration 3 of learning relationships among objects, we designed A11yBoard to support a two-finger flick gesture that reads out information about the closest object in the general flicking direction. By using flicks, a user can explore “radially” around an object to understand nearby objects in various directions. An example spoken report is: “Nearest object at 90 degrees, a black text at 50% from canvas left and 25% from canvas top, with size of 20% of canvas width and 30% of canvas height.”

Double-tap to traverse objects’ Z-order: Objects on the canvas are ordered along the Z-axis (i.e., their Z-order), which determines how objects would overlap each other if positioned in the same space. However, understanding object overlap and occlusion is a significant challenge for BLV users [60]. Therefore, A11yBoard supports using a double-tap gesture to traverse stepwise through objects’ by their Z-order, from top to bottom. (Recall that “step-in” and “step-out” sounds will have already informed users when they are inside multiple objects, indicating when objects are overlapping.) For example, if a text box is overlapping a rectangle, one double-tap would announce the text box, and a second double-tap would announce the rectangle. If a user double-taps again, A11yBoard cycles back to the top, announcing that it did so. By using Z-order traversals and “step-up”/“step-down” tones, A11yBoard reveals objects’ overlap and occlusion (see consideration 5 in Section 3).

4.2.2 Speech-Based Interactions. As described above, after an object has been selected via split-tap (see Figure 2b), when a user dwells with a finger on the screen, A11yBoard supports speech-based interactions to gain more information about it (see Figure 2c), including a variety of commands to access information and manipulate objects. A11yBoard announces “mic on”/“mic off” when the user’s finger dwells or lifts to start or stop “listening,” respectively. Below, we describe all speech commands in detail.

- **“Position”:** This command reports an object’s position, represented as percentage distances from the canvas’ top, left, bottom, and right edges to the object’s nearest edge. An example report is: “30% from canvas left, 40% from canvas top, 10% from canvas right, 20% from canvas bottom.” Individual commands like “left,” “top,” “right,” or “bottom” are also supported for single values.
- **“Size”:** This command reports an object’s size, similar to position, using a relative percentage of the canvas width and height. An example report is: “20% of canvas width, 10% of canvas height.” Similarly, “width,” “height” are also supported individually.

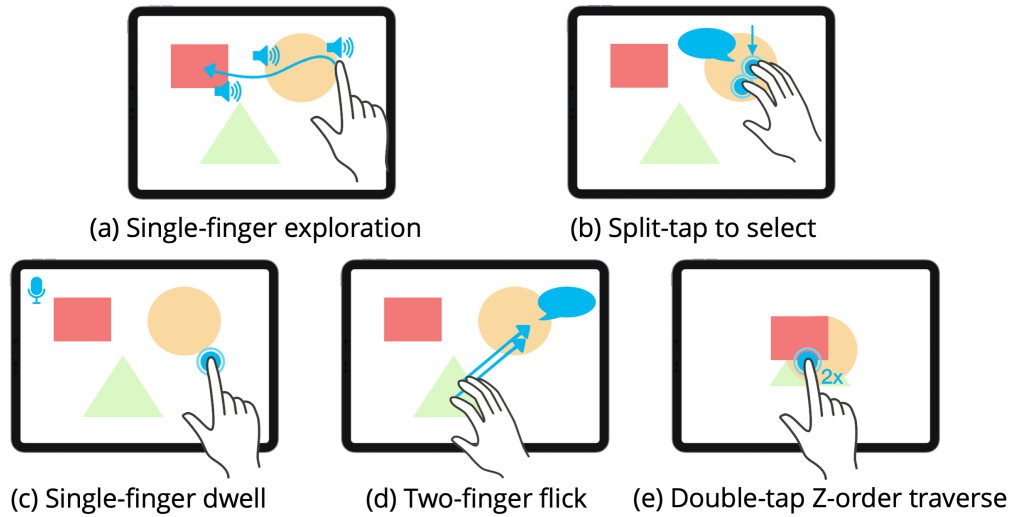


Figure 2: Five touch- and gesture-based interactions, including: (a) single-finger exploration to spatially “read” artboard objects, (b) split-tap to select an object and receive more information, (c) single-finger dwell to initiate speech recognition, (d) two-finger flick to reveal nearby objects in a given direction, and (e) double-tap to traverse stepwise in the Z-order of overlapping objects.

- “Color”: This command reports an object’s color name, such as “red” or “green” or “sky blue” or “goldenrod.” A11yBoard uses a public Python package called *webcolors* to map an object’s RGB (red, green, blue) values to its closest color name.
- “Text”: This command reads the text contents of any selected text box using text-to-speech.
- “Closest” and “Farthest”: These commands report the closest or farthest object, and its direction, from the selected object or current finger position. An example output is: “The closest object is a red circle to the south-southwest.” A11yBoard uses the closest named directions to report an object’s approximate direction. Furthermore, a number can be added after the commands to learn about a number of objects instead of only one. For example, “closest two” requests information about two objects that are closest to the finger’s position, reported in increasing distance. For simplicity, distances are calculated using the Euclidean distance from the finger to an object’s centroid. Together with two-finger flick to discover nearby objects, A11yBoard provides ways to learn about object relationships (see consideration 3 in Section 3).
- “Exact”: For all the speech commands above except “text,” the additional keyword “exact” can be appended to retrieve more precise information. For the “position” and “size” commands, adding “exact” changes the relative percentage numbers to exact pixel numbers. For the “color” command, RGB values will be reported instead of color names. For the “closest” and “farthest” commands, a specific integer degree (from 0-359 degrees counterclockwise from straight-right) is reported instead of named compass directions. Providing exact pixel, RGB, and degree information allows users to retrieve information when requested for fine-tuning objects and their relationships.

Besides using the above speech commands to retrieve detailed information about objects, A11yBoard’s speech commands can also be used to create or edit objects. (We refer to the former as “interpreting” an artboard whereas the latter is “authoring” an artboard.) When combined with touch and gesture interactions, A11yBoard supports creating, moving, and resizing objects as follows:

- “Create”: This command enables creation of different types of default objects under the finger where it dwells. The created object will be selected by default for further interactions. A11yBoard provides feedback via speech output to confirm the operation (e.g., “a rectangle has been created”).
- “Move”: After an object has been created or selected, this command enables the object to be moved. The moving process occurs in two phases. First, the “move” command is used to initialize the moving, and A11yBoard announces as much for confirmation (e.g., “a text box will be moved”). Second, users can continue exploring the artboard using the full set of touch, gesture, and speech interactions until they find a suitable movement destination. Once the user has found the desired location, they can say either “here” or “align” to complete the moving operation; “here” and “align” are described below. In contrast to traditional drag-and-drop, A11yBoard employs this two-phase object-movement design so that users can freely discover their moving destination without having to both find their object and also find their destination before issuing the command. Allowing users to *defer* the destination decision reduces cognitive load when moving objects.
- “Resize”: Similarly, this command is used to initialize object resizing, which must be accompanied by a corner or edge name from which to carry out the resizing (e.g., “top-left” or “right”). Like with “move,” resizing is separated into two phases, where users can freely explore the artboard after

initializing “resize” until they find a suitable place to resize the object to, or until they find another object to align with.

- “Here”: After users initiate moving or resizing processes, they can use “here” to move or resize the object to their current finger position. In the case of “move,” an object will have its center located at the finger position when “here” is said. In the case of “resize,” an object will have its designated corner or edge located at the finger position. A11yBoard also uses speech output to confirm the operation (e.g., “a text box has been moved to 35% from the canvas top and 25% from the canvas left”).
- “Align”: Similar to “here,” this command accompanies object moving and resizing. A11yBoard supports aligning the current object to another object’s edges. For example, when a user tries to align a paragraph text box’s left edge to the title text box’s left edge, they first select the paragraph text box and use the speech command “move.” Next, they use their reading finger to find the title box, split-tap to select it, and use the speech command “align left to left.” Speech output then confirms the operation (e.g., “a text box’s left edge has been aligned to a text box’s left edge”).
- “Cancel”: This command cancels a moving or resizing process underway, with speech output as feedback (e.g., “moving canceled”).

4.2.3 Universal Command Line. Although interactions through touch, gesture, and speech commands are powerful for interacting with 2-D objects, they are still limited for functions like changing discrete values of color, font family, and font size. Simple touches and gestures offer relatively limited inputs and having too many speech commands risks overloading the user. Therefore, to support additional commands, especially discrete operations, we created a “Universal Command Line,” a search-driven desktop keyboard interface (see Figure 3) equipped with speech output into which users can type commands like “color” or “font,” and then select from search results appearing in a drop-down menu made accessible by speech output. Users need not remember keywords for this command line, they need only to type a few characters related to their command. For example, a user might type, “Time” wishing to change a text box’s font to “Times New Roman.” Additional commands like “copy,” “paste,” “delete,” “bring to front,” and “send to back” are also supported for users to further edit objects. Finally, a vital keyboard capability in A11yBoard is that it also supports multiple undo and redo with the Ctrl+Z and Ctrl+Y key combinations.

5 PRELIMINARY STUDY SESSIONS

Prior to conducting a formal task-based usability evaluation, we conducted a series of informal iterative and participatory preliminary study sessions with three blind participants to elicit ideas and understand how *A11yBoard*’s developing combination of input/output modalities might work for them. These preliminary sessions provided important formative feedback in the development of A11yBoard.

5.1 Method

We recruited three participants via local email lists and word-of-mouth. One participant was a man and two participants were

women. Two participants were aged 25–34, and one participant was aged 55–64. The participants all had experience using digital artboard tools like Microsoft PowerPoint or Google Slides to make or use presentation slides. All reported considerable difficulty and frustration using such tools with conventional screen readers, affirming the need for a solution like A11yBoard.

In our informal sessions, we discussed A11yBoard’s features as they existed at the time. We asked participants to try those features and provide feedback. We also asked participants to suggest modifications to existing features or even new features entirely. We encouraged participants to be imaginative, freeing them from system constraints or implementation challenges.

We also asked participants to use A11yBoard to interpret an existing artboard and to edit objects within it. Along the way, we progressively introduced A11yBoard’s features and asked participants to try them to interpret or edit artboard objects. Participants conversed with us freely, and were encouraged to be critical of interactions they did not like. Each session lasted about 60–90 minutes and provided substantial input to our design iterations.

5.2 Results and Design Iterations

In general, participants were enthusiastic about A11yBoard, all enjoying using it to access the given artboard and using the interaction techniques to edit it. For example, all three participants were able to tell accurately what the artboard was generally about (e.g., a slide with a title in the top-left corner, a rectangle below that title, and a few other shapes on the right side). One participant particularly appreciated the separate two-phase design for moving and resizing objects, which allowed her to explore the destination first before “rushing into any decisions.” (As it happens, it also matches how Apple iOS’ VoiceOver enables moving application icons on its home screen, something a participant pointed out to us.) Participants also seemed to embrace A11yBoard’s multimodal interactions, felt that these modalities were natural and intuitive, and also gave useful suggestions for their improvement:

1. Provide an artboard overview: In our initial design, we only provided risk-free finger-reading to spatially explore artboard contents. However, all participants felt like they needed an “overview” command to receive summarized information of the entire artboard. To this end, we added an “overview” speech command in A11yBoard to summarize an artboard’s contents. For this command, we sort all objects from the canvas’ top-left corner to its bottom-right corner based on objects’ centroids’ distances from this diagonal, then read them out one by one using their color and type (e.g., “red rectangle”). The result is a rough top-left to bottom-right reading of objects on the canvas.

2. Tolerate accidental “gestures”: A recurring issue with our preliminary study participants was that when they tried to touch the screen to perform gestures, the actual and expected distances between their finger tips and the screen were mismatched. This would result, for example, in small finger movements when performing single-finger dwelling on the screen to initiate speech recognition. A11yBoard initially recognized this movement as an attempt to move the finger and read other objects, and any current object selection would be lost. To solve this usability issue, we improved the tolerance of A11yBoard’s gesture recognizer. If the

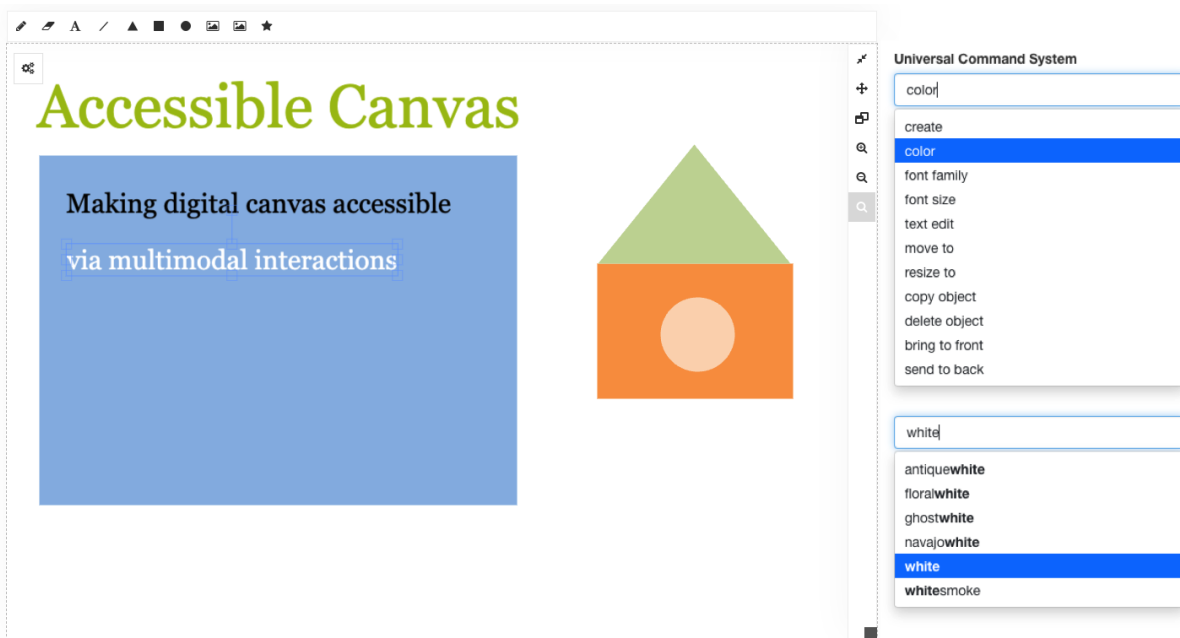


Figure 3: An example of using A11yBoard’s *Universal Command Line*. In the top text box, the user can choose from a variety of commands or type to search. In this example, the user went down the list and chose “color.” In the bottom text box, A11yBoard lists a series of supported colors for selection after the user typed “white.”

finger dwells on the screen within a small range, we consider that users are trying to initiate speech recognition (i.e., “mic on”). We also improved the system by tolerating off-screen gestures when the finger accidentally swipes off (and perhaps back onto) the touch screen.

3. Support increasing object familiarity: When accessing an existing artboard, users’ familiarity with the artboard should increase over time. One participant mentioned that after accessing the same set of objects for a while, they became familiar with those objects, and no longer wanted to hear the same detailed spoken reports every time they moved across the object. To support this, we abruptly cut any spoken object report when the reading finger moves out of the object or lifts from the screen. To make this even clearer, we added a short audio tone after cutting off spoken reports to indicate that the audio was intentionally cut.

4. Provide for inputting exact positions and sizes: One participant mentioned that although interpreting and editing objects’ positions and sizes on the touch screen device were intuitive and natural, they still wanted a way to accurately input an object’s size or position. For example, if a user wants to resize a rectangle precisely to a size of 102×101 pixels, it is difficult to accomplish this task using only a touch screen. Therefore, we added two keyboard commands to the Universal Command Line: “move to” and “resize to” for directly inputting an object’s position and size in pixels.

5. Include a “help” command: One participant felt they needed a “help” command when first learning A11yBoard. Therefore, we added a “help” speech command prefix to explain any of the other speech commands. For example, “help move” would be followed by an explanation of how to move an object: “Select an object and

say ‘move,’ then return to normal exploration until you find your destination, and say ‘here’ or ‘align’ an edge to an edge.”

Our preliminary session results indicated the usefulness of combining touch, gesture, speech, audio, and keyboard search for improving digital artboard accessibility. After iterating on A11yBoard through these preliminary study sessions, we conducted a formal task-based usability evaluation of A11yBoard, described below.

6 USER STUDY METHOD

The goal of our formal user study of *A11yBoard* was to evaluate our system’s most up-to-date version, replete with improvements from our preliminary study sessions, with a larger set of blind and low-vision (BLV) participants. We wanted to understand whether and how A11yBoard makes 2-D digital artboards accessible for both interpreting and editing tasks. To this end, we adopted a task-based usability study methodology [51]. We incorporated tasks from prior work that examined artboard accessibility with conventional screen readers [60], giving us a baseline for comparison. We adjusted some tasks as necessary to fit our specific study needs.

6.1 Participants

We recruited eight blind participants (Table 1) from the National Federation of the Blind (NFB)’s research program. The NFB spread the study invitation via a local email list. We also recruited some participants via snowball sampling. Four participants self-identified as men and four as women. Seven participants were blind and one participant had low-vision. Participants ranged in age from 18 – 64, with most between 25 – 44. Seven participants had previous

experience using digital artboard tools like Microsoft PowerPoint, Google Slides, or Apple Keynote to either read or (try to) create presentations, often with great difficulty. One participant (P8) had not used any such tools before, but expressed an interest in using digital artboards if they could be made made accessible. Two participants (P1, P2) had participated in our preliminary study sessions. Participants were compensated \$30-45 for the study session, which lasted about 60 – 90 minutes.

6.2 Apparatus

The tested apparatus for the study was the A11yBoard system described in Section 4 and iterated through preliminary study sessions described in Section 5. We built a server written in Python Flask to handle HTTP communications between the drawing tool, which appeared on a Macbook Pro laptop, and the touch screen application, which appeared on an Apple iPad Mini. The Macbook Pro showed the A11yBoard canvas in the Chrome web browser, maximized to full screen. The Macbook Pro had a 16" screen with 3072×1920 resolution. The iPad Mini had an 8.3" screen with 2266×1488 resolution. The iPad was used in landscape orientation.

Although most participants used Microsoft Windows, in our study, participants were not required to perform any general usage of the Macbook Pro laptop or iPad Mini device. Rather, they were confined only to using A11yBoard on both devices, and the only aspect of the Macbook Pro they used was the hardware keyboard with our *Universal Command Line* (see Section 4.2.3).

6.3 Procedure

The study was conducted in-person, individually with each of our eight participants, following all COVID-19 safety protocols. We brought the laptop and touch screen device to the location each participant preferred. Prior to the study session, each participant answered a few demographics questions via an online form and gave their consent in writing.

Each study session consisted of two halves, one focused on “interpretive tasks” and the other focused on “generative tasks” [60]. The former are tasks involved in understanding pre-existing artboards whereas the latter are tasks involved in creating and modifying artboard contents. In each session, we gave participants a tutorial on A11yBoard’s features lasting about 10-15 minutes, followed by a series of tasks where we recorded participants’ completion successes (or failures), breakdowns, workarounds, and any relevant think-aloud data that emerged. We did not record task times because times could be substantially affected by participants’ think-aloud behaviors, but all tasks, if considered successful, were completed in a reasonable amount of time.

6.3.1 Interpretive Tasks. For the half-session involving interpretive tasks, we asked participants to complete the following tasks on an artboard prepared in advance (see Figure 4). These tasks were adapted from prior work [60], including eight questions about describing the artboard in general; finding out about an object in a certain place; investigating its color, size, and position; finding out about a nearby object and its attributes; finding out about an object in a certain direction; finding out about an object stack; and finding out about objects occluded underneath.



Figure 4: The artboard used for interpretive tasks, including a title text box, a paragraph text box with a rectangle below, as well as several overlapping shapes on the right.

6.3.2 Generative Tasks. Similarly, for the half-session involving generative tasks, we made a blank artboard for creating and editing objects, and asked participants to complete a new set of tasks, including seven questions about creating a text box; typing text inside; creating another rectangle; moving and aligning it with another object; resizing the rectangle; duplicating and removing an object; and moving an object in Z-order (bring to top).

Participants consistently gave feedback during their performance of both interpretive and generative tasks, and we solicited further feedback at the end of each study session. We also reviewed all features used by a participant in their session and asked them how they felt about that feature, and how they might want to improve it, if at all. We also assessed participants’ perceived workload using the NASA Task Load Index questionnaire (NASA-TLX) [27]. Finally, we asked participants whether, if it were possible, they would allow A11yBoard to be integrated into their existing tools, doing our best to prompt participants to consider actual adoption into their lives.

6.4 Analysis

To analyze the data and observational results from our usability testing, we followed the standard procedures of described by Nielsen [51]. We employed think-aloud protocol, assigned tasks and observed task completions, breakdowns, and workarounds. We also observed how participants reacted to A11yBoard’s features. There was no need for formal coding of qualitative data because task measurements were objective counts and measures [51].

7 STUDY RESULTS

We organize our findings from our formal usability tests and post-test interviews in four parts: (1) overall impressions and performance, (2) interpretive task outcomes, (3) generative task outcomes, and (4) perceived workload.

Participant ID	Age Range	Gender	Vision	Mobile Device	Computer Device	Has Artboard Experience
P1	55-64	Woman	Blind	iPhone	Windows	Yes
P2	25-34	Woman	Blind	iPhone	Windows & Mac	Yes
P3	25-34	Woman	Blind	iPhone	Windows	Yes
P4	18-24	Man	Blind	iPhone	Windows	Yes
P5	25-34	Man	Blind	iPhone	Windows & Mac	Yes
P6	35-44	Man	Blind	iPhone	Mac	Yes
P7	35-44	Woman	Low-Vision	Android	Windows	Yes
P8	35-44	Man	Blind	iPhone	Windows	No but has interest

Table 1: A11yBoard formal study participants.

7.1 Overall Impressions and Performance with A11yBoard

All eight participants said that they had read presentation slides in their academic or professional lives. Textual content, such as slide titles and bullet points, were considered reasonably easy to access via a conventional screen reader, but slides with objects, images, arrows, diagrams, and so on were much less accessible. All participants had experienced barriers when using digital artboard drawing tools like Microsoft PowerPoint or Google Slides. In general, participants all reported that the contents inside a 2-D canvas are often too overwhelming to interpret, let alone manipulate.

All eight participants also expressed that A11yBoard’s combination of devices and multimodal interaction techniques were easily learned and used. Especially popular was the use of a touch screen device to explore the canvas via touch and gesture. Participants reported that this gave them a good spatial understanding of objects and their sizes and positions, which is not something BLV participants experience when using existing drawing tools and screen readers. During her interview, P1 said, “I really *really* like it, because, you know, using your fingers, I am a really tactile person, and it just makes a lot of sense to me. All these good spatial senses, they help a lot.” P5 also mentioned, “it was really not something [only] virtual; I can literally feel objects under my finger using those audio tones you set.”

Our BLV participants also had success in completing interpretive tasks using A11yBoard. Following prior work [60], we divided our participants’ success in completing tasks into five gradations: (1) complete success; (2) partial success; (3) the participant thought they were successful, but they were not; (4) the participant quit or was unable to finish the task; and (5) the participant did not attempt the task. For example, a task was noted as a “partial success” when participants either completed the task with aid of a hint, or completed only part of the task successfully.

After undertaking the tasks, participants gave feedback on how A11yBoard could be improved. Below, we report participants’ suggestions in detail, in hopes they might inform not only A11yBoard’s future development, but the development of related systems as well.

7.2 Interpretive Task Outcomes

In this subsection, we report on the outcomes of participants’ interpretive tasks, including comparisons to commercial screen readers [60]. We also report on participants’ perceptions of task load and their suggestions for how to improve A11yBoard.

7.2.1 Task Performance. First, study participants completed most attempted tasks with some success (Figure 5). In fact, four tasks had 100% success. Specifically, using single-finger exploration with audio feedback was highly successful, as all participants were able to understand where they were on the canvas, where objects were on the canvas, and how big those objects were (task 2). P1 and P5 specifically mentioned that single-finger exploration was intuitive for understanding an object’s properties. For understanding more complex scenarios like two objects overlapping, participants also liked how consecutive audio sounds indicated object overlap (tasks 5, 7). Note that P7’s task 7 was marked as “did not attempt,” and P5’s was marked as having “partial success,” which was because P5 had difficulty in locating the stacked objects on the right side of the artboard until given a hint.

Second, using speech commands (see Section 4.2.2), all participants were able to request an overview of the artboard (task 1), followed by detailed object properties like positions, sizes, and colors (tasks 3, 4). However, in tasks 3 and 4, some participants (P2, P3, P8) were marked as having “partial success” because they had some difficulty entering the speech interaction mode via single-finger dwelling. Despite our improvements from preliminary testing, for some participants, it remained challenging to estimate how far their fingers were above the screen. Therefore, when their fingers landed on the screen, they slid a short distance. Although we iterated the A11yBoard implementation to tolerate such accidental sliding gestures, and indeed this helped, the issue was not completely resolved for these three participants who had greater-than-expected finger drift. Eventually, P2 and P3 managed to overcome this issue by placing the iPad Mini on their laps instead of on the desk, giving them better control. Similarly, P8 overcame the issue after drying his fingertip, successfully completing tasks 3 and 4. Nevertheless, tolerance for accidental gestures remains an important usability issue. The ability to set a personalized tolerance might be useful here.

Third, all participants also completed task 6 successfully using the two-finger flick gesture to discover nearby objects in a specific direction. They also used double-tap to access objects occluded beneath other objects (task 8). (Due to time limitations, one participant (P6) did not attempt to do task 8, but all who attempted it were successful.)

7.2.2 Comparing to Conventional Screen Readers. Because we based our tasks on those from prior work studying conventional screen

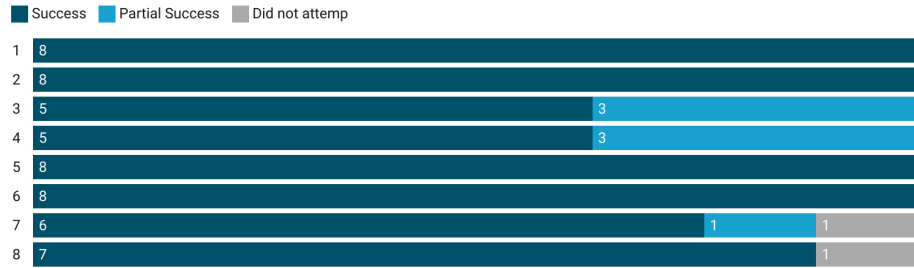


Figure 5: Task completion outcomes for interpretive tasks 1-8. The tasks are listed in Section 6.3.1. All attempted tasks were at least partially successful.

readers’ interpretations of commercial artboards [60], we can compare our task success rates to those. This is admittedly a coarse comparison, as the two studies were not identical. Nevertheless, our results showed that A11yBoard has great potential for improving the completion of interpretive tasks over conventional screen readers. For example, using a conventional screen reader, participants in prior work were only 25%, 58%, and 50% successful or partially successful at identifying an object’s color, width, and position on an artboard, respectively. Using A11yBoard, 100% of our participants successfully identified these properties of objects in similar tasks using both single-finger exploration and speech-based interactions. For determining objects’ relative positions (e.g., how objects are directionally situated around one another), only 36% of past participants were successful or partially successful. With A11yBoard, this understanding was easily achieved, and tasks 5 and 6 were 100% successful. In addition, unlike for multiple participants using conventional screen readers, with A11yBoard there were no “quit or was unable to finish” or “thought to be a success” outcomes for any session in our study. Importantly, because A11yBoard provided feedback for whether any operation was successful after being attempted, our participants never completed a task thinking they had been successful when they had not. This improvement alone represents a significant advance over artboard interaction with conventional screen readers.

7.2.3 Participants’ Feedback. Along with the task completion outcomes above, we captured participants’ feedback on A11yBoard and their suggestions for its improvement. Here we report on this feedback, grouped by features.

Reactions to single-finger exploration and split-tap selection. Seven of eight participants liked using the combination of single-finger exploration and split-tap selection. P6 thought that A11yBoard’s touch and gesture features had the potential to be used in other 2-D workspaces, like on maps, graphs, or diagrams. When exploring objects’ positions, P4’s strategy was to repeatedly use his finger to cross the four edges of an object’s bounding box multiple times, “feeling” the object’s dimensions on the canvas. Based on this strategy, P4 mentioned that they wished “the app could tell me which edge I am entering.” Meanwhile, P2 and P4 found an alternative to performing a one-handed split-tap for object selection, which was to use another hand’s finger to split-tap (see Figure 2b).

Reactions to audio feedback. Seven of eight participants liked the “step-up” and “step-down” audio sounds to indicate crossing

into and out of an object. This design was also generally successful at informing participants when objects were overlapping. However, if the overlap was significant or occlusion was total, some confusion could arise. To this end, P4 mentioned, “it would be better if the audio pitch [grew] higher and higher as you step into overlapped objects.” We had not considered augmenting the simple audio “step-in” and “step-out” sounds with pitch to convey extra information.

Reactions to speech output. A11yBoard provides feedback in the form of speech output for most commands. For example, various commands elicit percentage or pixel values, color names or RGB values, text and font properties, compass direction names or degree values, and command confirmations. Six of eight participants appreciated the different speech output reports and found them informative. However, two of eight participants expressed that they thought the “exact” outputs (pixels, RGB values, font properties, and degree values) were sometimes hard to interpret. P3 took awhile to understand what bounding boxes and their corresponding four values for top, left, bottom, and right edges meant, and said such values were hard for people who are not “math focused.” After we explained further, P3 mentioned that they would prefer to have all values reported in centimeters or inches instead of pixels, which presents an opportunity for personalization. Along those lines, seven of eight participants mentioned that they would like a *customization page* to set their preferences, including the levels of detail for reporting objects, how to report distances or degrees, and similar. P1 said, “The text is at north-northeast. I love it! You know, this is one of the things that individuals should be able to set up, because I love the ‘north-northeast,’ that works for me, but for some people, they might prefer degrees. For me, I [would] have to think about degrees.”

7.3 Generative Task Outcomes

Below we report on the outcomes of participants’ generative tasks, including their task completion success and how participants’ performance with A11yBoard compared to that of conventional screen readers from prior work [60]. We also discuss participants’ suggestions for improving A11yBoard.

7.3.1 Task Performance. First, participants completed 47 of 49 attempted tasks with partial or total success (Figure 6). Two tasks (9, 11) had 100% success rates, which meant that all participants managed to use speech-based interactions to create objects. P7 said,

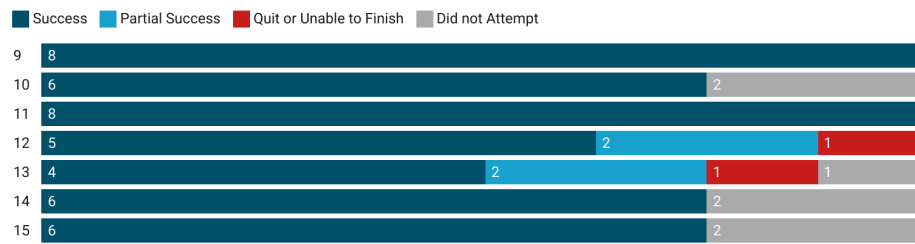


Figure 6: Task completion outcomes for generative tasks 9-15. The tasks are listed in Section 6.3.2. Only 2 of 49 attempted tasks were unsuccessful, with one participant unable to finish task 12 and one participant unable to finish task 13.

“it [creating an object] was perfect, a simple but powerful command, and I don’t need to remember anything else, unlike all the gestures.”

Second, after participants created a few objects on the artboard, we asked them to move a text box to the top-left corner as a title, and align with it a rectangle to serve as a paragraph (tasks 12, 13). Four of eight participants reported that moving and resizing were challenging, because these maneuvers require using a series of input and output modalities, including a speech command that requires additional parameters (e.g., “resize top-left”), followed by single-finger exploration and another speech command with more parameters (e.g., “align left to left”). Nevertheless, seven of eight participants were successful at moving tasks and six of eight participants were successful at resizing tasks. P8 was unable to finish the moving and resizing task because of a constant background noise in his preferred study location, which caused the speech recognition to fail. P5 declined to attempt the resizing task because he thought it was similar to moving and wanted to finish sooner. P3’s and P6’s moving and resizing tasks were marked as “partial success” because they asked researchers to help them locate artboard objects after becoming distracted.

Third, for using keyboard-based commands to change text, font, color, and perform other operations (tasks 10, 14, 15), all participants felt that the Universal Command Line was simple and straightforward. Two participants (P4 and P8) did not attempt to use keyboard commands due to study time limits.

7.3.2 Comparing to Conventional Screen Readers. We compared our generative task outcomes to similar tasks performed with conventional screen readers as reported in prior work [60]. With conventional screen readers, tasks related to creating, moving, and resizing shapes all had success rates under 50%. In contrast, with A11yBoard, almost all participants who attempted the tasks successfully created, moved, and resized objects. Changing text or shape color fared reasonably well in prior work (reported to be 60% and 45.5% respectively), but were 100% successful in A11yBoard due to its Universal Command Line interface.

7.3.3 Participants’ Feedback. Below we report on participants’ feedback and suggestions for improvement, grouped by feature.

Reactions to creating objects. Five of eight participants liked to use speech commands to create objects. However, some participants (P3, P4, P8) mentioned that besides speech commands, they would prefer to have gestures for creating objects. After completing the creation tasks, P4 thought that using a speech command to

create an object was time-consuming, and sometimes he would prefer to “have something fast, personally.” P4 gave an example of using a rotor like in VoiceOver, where he can move two fingers as if he is turning a dial. Each rotor option represents an object type, and lifting a finger means choosing to create that object. Other participants expressed their preference for placing an object under their finger instead of at the center of the screen. P3 also proposed an idea of using a finger to draw a shape. For example, if a user performs a two-finger dwell, they could be prompted with a spoken prompt to draw their desired object with a finger on the screen.

P5 mentioned that he would like to use AI-generated design ideas to automatically create artboards, like slide templates. In this case, BLV users would only need to input text or images they want to put on an artboard, and let an AI “designer” produce an aesthetic slide. In such a scenario, an A11yBoard user could then interpret the AI designer’s slides and make changes if desired.

Reactions to moving and resizing. While some participants reported that moving and resizing were challenging (P6: “It takes a lot of training”), all participants appreciated A11yBoard’s design separating “initializing” and “finalizing” moving or resizing. P2 compared it with how Apple’s VoiceOver currently supports moving app icons on the main screen—first choose an app to move, and then “drop” the app at the destination. A11yBoard employs the same approach. Sighted designers can drag-and-drop objects to a desired place using visual confirmation, but for BLV users, it can be challenging to learn about all objects they encounter along an exploratory movement path. Therefore, the drag-and-drop style of moving and resizing increases users’ cognitive load. P4 also suggested optimizing the moving and resizing procedure by adding a double-tap at the end to “drop” the object at the destination.

P3 pointed out that although A11yBoard currently supports resizing and moving to align one object with another, it does not support aligning with another object with a gap between, like aligning a rectangle one inch below a text box. P3 suggested that we should combine A11yBoard’s speech and keyboard modalities to make this available. However, P3 also thought that it added to the complexity of existing interactions.

Reactions to keyboard commands. Although participants felt that using keyboard commands was straightforward, two participants reported that they would lose track of the object they selected after switching to keyboard input. After using the keyboard, they had to re-locate the object of interest. Therefore, they suggested that we add some navigational help to move the user’s reading

finger toward the currently selected object. Access Overlays [35] employed a similar tactic by using speech output to guide a user's finger to desired screen content.

7.4 Example User-Generated Artboard

Figure 7 shows an example of a user-generated artboard (P7). After P7 completed her generative tasks, she wanted to produce an artboard of her own. We gave P7 the instruction to put a title and a paragraph on the left side and align their left and right edges, and then draw any shape she preferred. P7 chose to draw a house by using a triangle and a rectangle. P7 first created a rectangle using the speech command “create rectangle,” and then moved it using the “move here” command. Then P7 created a triangle and resized it using the keyboard command “resize,” a keyboard command we added after our preliminary study sessions. Finally, P7 aligned the triangle and the rectangle using the “align” command.

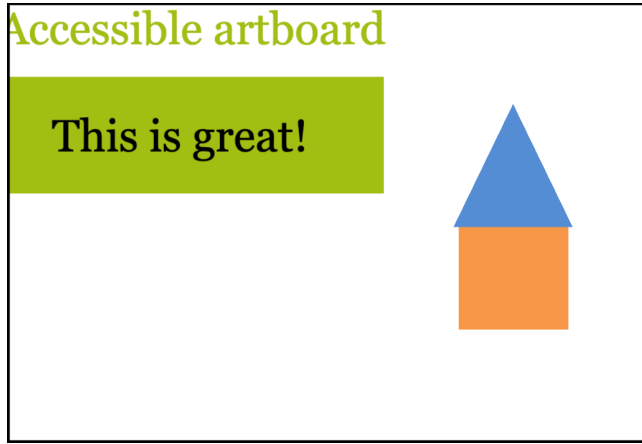


Figure 7: An artboard created by P7 in her generative tasks. Note that the slightly cut-off “A” at the top-left corner occurred because P7 created a text box at the top-left corner, and as the box’s centroid was placed under P7’s finger by default, its top-left corner was placed outside of the canvas.

7.5 Subjective Workload Ratings

We used the NASA TLX questionnaire [26, 27] to collect perceived workload ratings for A11yBoard. The NASA TLX instrument asks participants to rate the workload of a task on six scales: mental demand, physical demand, temporal demand, performance, effort, and frustration. Each scale ranges from low (1) to high (20) except performance, which ranges from perfect (1) to failure (20). Thus, for all scales, lower is better and corresponds to less perceived workload. We also classified a workload score x into four categories: low ($1 \leq x < 6$), somewhat low ($6 \leq x < 11$), somewhat high ($11 \leq x < 16$), and high ($16 \leq x \leq 20$). We asked participants to rate interpretive tasks (Figure 8) and generative tasks (Figure 9).

For interpretive tasks, our results indicate that A11yBoard required low mental ($M = 3.1, SD = 1.8$), physical ($M = 3.4, SD = 1.3$), and temporal ($M = 3.6, SD = 3.5$) demand, and has high perceived performance ($M = 3.4, SD = 1.3$), somewhat low effort ($M = 6.6, SD = 3.0$), and low frustration ($M = 2.6, SD = 1.9$).

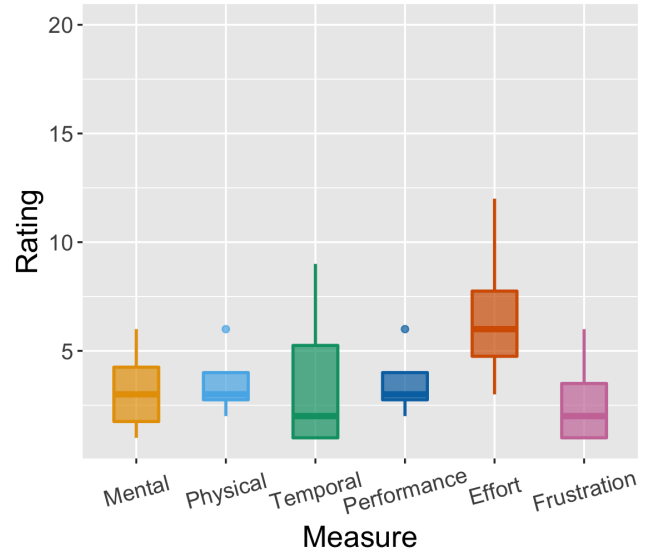


Figure 8: Box plots of NASA TLX perceived workload ratings for interpretive tasks. Scales ranged from 1-20. On all scales, lower is better, corresponding to a lesser workload.

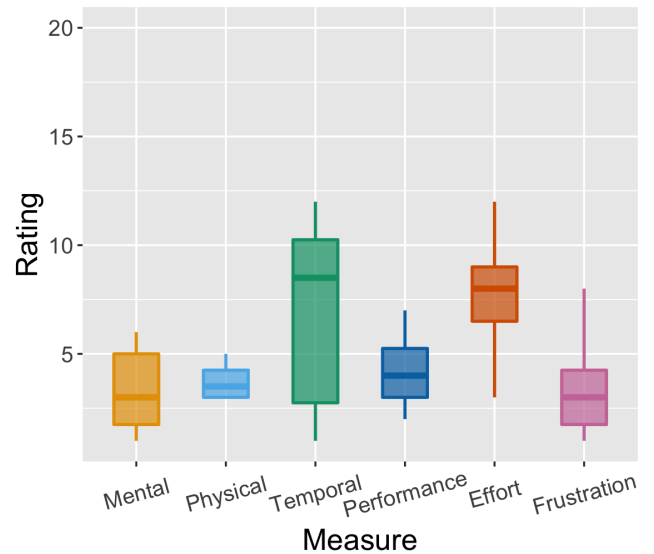


Figure 9: Box plots of NASA TLX perceived workload ratings for generative tasks. Scales ranged from 1-20. On all scales, lower is better, corresponding to a lesser workload.

For generative tasks, our results indicate that A11yBoard required low mental ($M = 3.3, SD = 2.0$) and physical ($M = 3.8, SD = 0.9$) demand, and somewhat low temporal demand ($M = 7.0, SD = 4.3$). Results also indicate that A11yBoard had high perceived performance ($M = 4.3, SD = 1.7$), somewhat low effort ($M = 7.6, SD = 2.7$), and low frustration ($M = 3.4, SD = 2.3$).

Thus, A11yBoard's workload ratings were all low or somewhat low, indicating workload was not a concern as it seemed to be with conventional screen readers [60].

8 DISCUSSION

In this section, we zoom out to offer a summary of results and feedback about A11yBoard from our user study participants. We also discuss A11yBoard's limitations and avenues for future work in this research effort.

8.1 Summary of Results and Feedback

To summarize our results and feedback from the study, we first revisit the overall task success rates in comparison to prior work. We then talk about summarized benefits and participants' suggestions for improving A11yBoard.

8.1.1 A11yBoard's Task Success. Our eight participants had considerable success in accomplishing both interpretive and generative tasks with A11yBoard. For interpretive tasks, in all, participants attempted 62 tasks and had at least "partial success" with all of them. For generative tasks, in all, participants attempted 49 tasks and had at least "partial success" with 47 of them (96%). Both of these outcomes compare very favorably to prior work [60] showing how BLV participants struggled with artboards using conventional screen readers.

8.1.2 A11yBoard's Benefits. From our post-session interviews, we provide a summary of A11yBoard's benefits below. These benefits can also serve as design guidelines for future researchers to consider when designing accessible 2-D creative spaces.

Intuitive spatial reasoning. Participants liked the touch screen-based finger-reading experience for exploring their artboard canvas and its objects. The absolute mapping of the artboard from the laptop display to the iPad Mini gave participants a directness of engagement that is currently missing when interacting with artboards only on a desktop or laptop computer. Also, our "step-up" and "step-down" sounds, simple as they were, provided quick spatial awareness of objects and their relationships, supporting efficient canvas exploration.

Multimodal access to objects' properties and relationships. Participants also enjoyed the richness of A11yBoard's multimodal interactions for accessing object information in a risk-free way. Specifically, participants felt a great deal of control with all modalities combined. The search-based Universal Command Line keyboard feature also allowed participants to perform other actions if desired. Finally, participants also liked how A11yBoard provided different levels of detail when describing objects.

Eyes-free object creation and editing. Beyond just accessing an existing artboard, participants could, for the most part, successfully create and edit an artboard of their own. They especially appreciated how we designed creation and editing features, keeping them separate from interpreting features such that the fear of "making a mistake" or "messing something up" while exploring was minimized. Multiple participants explicitly said that they saw the "good intention" behind A11yBoard in its design for BLV users based on their perspectives and input, avoiding overwhelming them with all the richness possible in an artboard's 2-D information space.

8.1.3 Suggestions for Improvement. In the aforementioned study results, we reported some participants' ideas for improving A11yBoard. Here, we summarize three higher-level possibilities for improving A11yBoard that were raised by participants. We also consider the summarized suggestions as part of design guidelines for future researchers to include when designing such tools.

Customization of reports. Most participants mentioned that they would like to have a "settings page" to be able to modify aspects of A11yBoard, such as units of measurement, verbosity of speech output, accidental "gesture" tolerance, and so on. Participants expressed a range of preferences for how objects' properties should be reported and in how much detail. We agree that this flexibility is important for any accessible system, in keeping with principles of ability-based design [71, 72], and suggest that a high degree of personalization be upheld as a design guideline for A11yBoard and other accessible creativity tools.

Consider AI assistance. Participants mentioned several possibilities of using AI as an assistant for identifying shapes drawn by users and creating templated artboards from the beginning, at least for when artboards are used for presentations. We recognize that accessibility is a driving force for AI solutions, and applying AI in assistive technologies is a promising method for developing intelligent content creation tools for people with disabilities.

Deeper integration of modalities. Although A11yBoard is a multimodal multi-device interactive system, most features use a single modality (e.g., keyboard input) or combine two modalities in a simple way (e.g., touch and speech). Participants suggested that A11yBoard could potentially support richer interactions by integrating modalities together. For example, by combining touch (for exploration), gesture (for enabling speech interaction), speech (for voice commands), and keyboard (for inputting a specific number), A11yBoard could support complex moving and resizing to a specific place (e.g., one inch below another object). A tradeoff here is to keep any additional complexity at a minimum, allowing simple, if multiple, ways of accomplishing common tasks.

8.2 Limitations

As described in Section 4, A11yBoard is our heavily modified version of *DrawerJS* [61], an open-source drawing tool. Although *DrawerJS* provides most common artboard features for creating and editing objects, it is not a mature artboard tool like Microsoft PowerPoint or Google Slides. Therefore, it poses some significant limitations (e.g., it only supports limited shapes and text styles, and does not support some common keystrokes). Also, despite A11yBoard using a custom set of interactions that does not rely on existing screen readers, participants' prior knowledge of screen readers might still affect their perceptions and expectations of *any* screen-reading tool.

Another limitation is that our user study tasks, although carefully adapted from prior work [60], could not fully capture the complexity of real-world creative educational and professional work. The nature of a lab study limits the generalizability of our A11yBoard evaluation. That said, as a first step in system development, iterative informal study sessions and a formal usability study are both appropriate; in the future, a field deployment supporting actual work practices would be enlightening. At this stage, our research focus

was on perfecting our multimodal multi-device interactions, which were indeed useful and usable to participants, yet still with some room for improvement. These interactions might also be usable in other systems beyond A11yBoard. For example, rich formatted text documents containing images, tables, shapes, and charts might use our proposed interactions to make different visual elements accessible to BLV users.

8.3 Future Work

For a system with multimodal features, and an accessibility challenge as demanding and important as 2-D spaces, it is not surprising that there are multiple avenues for future work. In addition to addressing aforementioned limitations, there are strategic avenues for additional research in this space.

We plan to iterate upon A11yBoard and ultimately make it open source. Important additions to A11yBoard include making it customizable, as described above, and adding certain AI-assisted features, for example, for identifying and fitting shapes drawn by fingers. We also plan to create a self-guided tutorial to support users learning A11yBoard.

Beyond making these significant additions to A11yBoard, we plan to investigate whether and how A11yBoard might be made to work with existing technologies, including screen readers like VoiceOver, TalkBack, NVDA, and JAWS, and commercial slide-making tools like Microsoft PowerPoint, Google Slides, and Apple Keynote. Numerous technical limitations undoubtedly abound here, not the least of which is that extensions to these programs do not often give full programmatic access to artboard object information in these tools. Beyond creating application extensions, it is possible to work with designers and developers of these tools to incorporate some of A11yBoard's capabilities and interactions directly.

Finally, future work in making other creativity tools accessible is inspiring to us, such as for paint canvases (i.e., pixel-based drawing tools), wireframing tools, storyboarding tools, animation tools, video editing tools, or audio editing tools, to name a few. Undoubtedly, myriad challenges exist in each of these domains, but many of the techniques we incorporated in A11yBoard might be applicable to these tools as well. For example, allowing users to explore a 2-D information space with their finger using a touch screen device whose contents are mirrored on a desktop or laptop display seems to be a promising multimodal multi-device design approach for non-visual spatial interactions.

9 CONCLUSION

We have presented *A11yBoard*, a multimodal multi-device system for making digital artboards accessible to blind and low-vision (BLV) users. Our dual focus has been in making artboards accessible both for *interpretation* and *generation*. A fundamental challenge for A11yBoard was in escaping the tendency for most screen readers to assume a linear 1-D “information stream,” instead enabling a way for BLV users to directly interact with a rich 2-D “information space.” To achieve this, A11yBoard employs multimodal interaction techniques across multiple devices. Specifically, A11yBoard combines touch, gesture, audio, and speech output, voice commands via speech recognition, and keyboard commands via search, all in an effort to support multiple command pathways and multiple levels

of output detail. To ensure A11yBoard's usability and usefulness, it was designed through iterative participatory sessions with BLV users.

To formally assess the performance of A11yBoard, we conducted task-based usability tests and participant interviews with eight BLV users. A11yBoard significantly increased the success rate of both interpretive and generative tasks compared to artboard interaction via conventional screen readers [60]. In general, participants felt that A11yBoard provided: (1) intuitive spatial reasoning about two-dimensional objects, (2) ready access to objects' properties and relationships through multimodal interactions, and (3) eyes-free creating and editing of objects to establish their desired properties.

Going forward, it is our hope that A11yBoard can serve as a proof point showing how people with disabilities can be moved from mere *consumers* of content to *creators* of their own content. While it remains of utmost importance to ensure that content, products, and services are indeed accessible when used, we believe that a vital step is to empower all people to unleash their creative energies using accessible tools, limited only by their imaginations.

ACKNOWLEDGMENTS

This work was supported in part by the University of Washington Center for Research and Education on Accessible Technology and Experiences (CREATE). Any opinions, findings, conclusions or recommendations expressed in our work are those of the authors and do not necessarily reflect those of any supporter.

REFERENCES

- [1] Fernando Alonso, José L. Fuertes, Ángel L. González, and Loïc Martínez. 2008. User-interface modelling for blind users. In *International Conference on Computers for Handicapped Persons*. Springer, 789–796.
- [2] Apple. 2022-09-09. Accessibility - Vision. <https://www.apple.com/accessibility/vision/>
- [3] Amine Awada, Youssef Bou Issa, Clara Ghannam, Joe Tekli, and Richard Chbeir. 2012. Towards digital image accessibility for blind users via vibrating touch screen: A feasibility test protocol. In *2012 Eighth International Conference on Signal Image Technology and Internet Based Systems*. IEEE, 547–554.
- [4] Amine Awada, Youssef Bou Issa, Joe Tekli, and Richard Chbeir. 2013. Evaluation of touch screen vibration accessibility for blind users. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility*. Article 48, 2 pages.
- [5] Cynthia L. Bennett, Erin Brady, and Stacy M. Branham. 2018. Interdependence as a frame for assistive technology research and design. In *Proceedings of the 20th international acm sigaccess conference on computers and accessibility*. 161–173.
- [6] Jeff Bilmes, Xiao Li, Jonathan Malkin, Kelley Kilanski, Richard Wright, Katrin Kirchhoff, Amarnag Subramanya, Susumu Harada, James Landay, Patricia Dowden, and Howard Chizeck. 2005. The Vocal Joystick: A voice-based human-computer interface for individuals with motor impairments. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. 995–1002.
- [7] Paul Blenkhorn and D. Gareth Evans. 1998. Using speech and touch to enable blind people to access schematic diagrams. *Journal of Network and Computer Applications* 21, 1 (1998), 17–29.
- [8] Jens Bornschein and Gerhard Weber. 2017. Digital Drawing Tools for Blind Users: A State-of-the-Art and Requirement Analysis. In *Proceedings of the 10th International Conference on Pervasive Technologies Related to Assistive Environments (PETRA '17)*. Association for Computing Machinery, New York, NY, USA, 21–28. <https://doi.org/10.1145/3056540.3056542>
- [9] L. M. Brown, S. A. Brewster, S. A. Ramloll, R. Burton, and B. Riedel. 2003. Design guidelines for audio presentation of graphs and tables. <https://eprints.gla.ac.uk/3196/>
- [10] Tim Brown et al. 2008. Design thinking. *Harvard business review* 86, 6 (2008), 84.
- [11] William Buxton, Ralph Hill, and Peter Rowley. 1985. Issues and techniques in touch-sensitive tablet input. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*. 215–224.

- [12] Matt Calder, Robert F Cohen, Jessica Lanzoni, and Yun Xu. 2006. PLUMB: an interface for users who are blind to display, create, and modify graphs. In *Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility*. 263–264.
- [13] Robert F Cohen, Arthur Meacham, and Joelle Skaff. 2006. Teaching graphs to visually impaired students using an active auditory interface. *ACM SIGCSE Bulletin* 38, 1 (2006), 279–282.
- [14] Ádám Csapó, György Wersényi, Hunor Nagy, and Tony Stockman. 2015. A survey of assistive technologies and applications for blind users on mobile platforms: a review and foundation for research. *Journal on Multimodal User Interfaces* 9, 4 (2015), 275–286.
- [15] Franco Delogu, Massimiliano Palmiero, Stefano Federici, Catherine Plaisant, Haixia Zhao, and Olivetti Belardinelli. 2010. Non-visual exploration of geographic maps: does sonification help? *Disability and Rehabilitation: Assistive Technology* 5, 3 (2010), 164–174.
- [16] Julie Ducasse, Anke M Brock, and Christophe Jouffrais. 2018. Accessible interactive maps for visually impaired users. In *Mobility of visually impaired people*. Springer, 537–584.
- [17] Md Ehtesham-Ul-Haque, Syed Mostofa Monsur, and Syed Masum Billah. 2022. Grid-Coding: An Accessible, Efficient, and Structured Coding Paradigm for Blind and Low-Vision Programmers. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. Article 44, 21 pages.
- [18] Julie Fraser and Carl Gutwin. 2000. A framework of assistive pointers for low vision users. In *Proceedings of the fourth international ACM conference on Assistive technologies*. 9–16.
- [19] Nicholas A Giudice, Hari Prasath Palani, Eric Brenner, and Kevin M Kramer. 2012. Learning non-visual graphical information using a touch-based vibro-audio interface. In *Proceedings of the 14th international ACM SIGACCESS conference on Computers and accessibility*. 103–110.
- [20] Google. 2022-09-09. Speech-to-Text: Automatic Speech Recognition. <https://cloud.google.com/speech-to-text>
- [21] Google. 2022-09-09. Text-to-Speech: Lifelike Speech Synthesis. <https://cloud.google.com/text-to-speech>
- [22] Joan Greenbaum and Morten Kyng. 2020. *Design at work: Cooperative design of computer systems*. CRC Press.
- [23] William Grussenmeyer and Elke Folmer. 2016. AudioDraw: user preferences in non-visual diagram drawing for touchscreens. In *Proceedings of the 13th International Work for All Conference (W4A '16)*. Association for Computing Machinery, New York, NY, USA, Article 22, 8 pages. <https://doi.org/10.1145/2899475.2899483>
- [24] Tiago Guerreiro, Paulo Lagoa, Hugo Nicolau, Daniel Goncalves, and Joaquim A Jorge. 2009. From Tapping to Touching: Making Touch Screens Accessible to Blind Users (vol 15, pg 48, 2008). *IEEE MULTIMEDIA* 16, 1 (2009), 13–13.
- [25] Susumu Harada, Jacob O Wobbrock, and James A Landay. 2007. Voicedraw: a hands-free voice-driven drawing application for people with motor impairments. In *Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility*. 27–34.
- [26] Sandra G Hart. 2006. NASA-task load index (NASA-TLX); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting*, Vol. 50. Sage publications Sage CA: Los Angeles, CA, 904–908.
- [27] Sandra G Hart and Lowell E Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in psychology*. Vol. 52. Elsevier, 139–183.
- [28] Anthony J Hornof and Anna Cavender. 2005. EyeDraw: enabling children with severe motor impairments to draw with their eyes. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 161–170.
- [29] Deepak Jagdish, Rahul Sawhney, Mohit Gupta, and Shreyas Nangia. 2008. Sonic Grid: an auditory interface for the visually impaired to navigate GUI-based environments. In *Proceedings of the 13th international conference on Intelligent user interfaces*. 337–340.
- [30] JAWS. 2020-10-15. Jaws® – freedom scientific. <https://www.freedomscientific.com/products/software/jaws/>
- [31] Chandrika Jayant, Matt Renzelmann, Dana Wen, Satria Krisnandi, Richard Ladner, and Dan Comden. 2007. Automated tactile graphics translation: in the field. In *Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility*. 75–82.
- [32] Nikolaos Kaklanis, Konstantinos Votis, and Dimitrios Tzovaras. 2013. A mobile interactive maps application for a visually impaired audience. In *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*. Article 23, 2 pages.
- [33] Hesham M. Kamel and James A. Landay. 2000. A study of blind drawing practice: creating graphical information without the visual channel. In *Proceedings of the fourth international ACM conference on Assistive technologies (Assets '00)*. Association for Computing Machinery, New York, NY, USA, 34–41. <https://doi.org/10.1145/354324.354334>
- [34] Shaun K. Kane, Jeffrey P. Bigham, and Jacob O. Wobbrock. 2008. Slide rule: making mobile touch screens accessible to blind people using multi-touch interaction techniques. In *Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility (Assets '08)*. Association for Computing Machinery, New York, NY, USA, 73–80. <https://doi.org/10.1145/1414471.1414487>
- [35] Shaun K. Kane, Meredith Ringel Morris, Annuska Z. Perkins, Daniel Wigdor, Richard E. Ladner, and Jacob O. Wobbrock. 2011. Access overlays: improving non-visual access to large touch screens for blind users. In *Proceedings of the 24th annual ACM symposium on User interface software and technology - UIST '11*. ACM Press, Santa Barbara, California, USA, 273. <https://doi.org/10.1145/2047196.2047232>
- [36] Shaun K. Kane, Meredith Ringel Morris, and Jacob O. Wobbrock. 2013. Touchplates: low-cost tactile overlays for visually impaired touch screen users. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '13)*. Association for Computing Machinery, New York, NY, USA, Article 22, 8 pages. <https://doi.org/10.1145/2513383.2513442>
- [37] Shaun K Kane, Kristen Shinohara, and Jacob O Wobbrock. 2015. OneView: Enabling Collaboration Between Blind and Sighted Students. (2015).
- [38] Shaun K Kane, Jacob O Wobbrock, and Richard E Ladner. 2011. Usable gestures for blind people: understanding preference and performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 413–422.
- [39] Andrea R Kennel. 1996. Audiograf: A diagram-reader for the blind. In *Proceedings of the second annual ACM conference on Assistive technologies*. 51–56.
- [40] Martin Kurze. 1996. TDraw: a computer-based tactile drawing tool for blind people. In *Proceedings of the second annual ACM conference on Assistive technologies (Assets '96)*. Association for Computing Machinery, New York, NY, USA, 131–138. <https://doi.org/10.1145/228347.228368>
- [41] Martin Kurze. 1996. TDraw: a computer-based tactile drawing tool for blind people. In *Proceedings of the second annual ACM conference on Assistive technologies*. 131–138.
- [42] Richard E Ladner. 2015. Design for user empowerment. *interactions* 22, 2 (2015), 24–29.
- [43] Richard E Ladner, Melody Y Ivory, Rajesh Rao, Sheryl Burgstahler, Dan Comden, Sangyun Hahn, Matthew Renzelmann, Satria Krisnandi, Mahalakshmi Ramasamy, Beverly Slabosky, Andrew Martin, Amelia Lacenski, Stuart Olsen, and Dmitri Groce. 2005. Automating tactile graphics translation. In *Proceedings of the 7th international ACM SIGACCESS conference on Computers and accessibility*. 150–157.
- [44] Steven Landau and Lesley Wells. 2003. Merging tactile sensory input and audio data by means of the Talking Tactile Tablet. In *Proceedings of EuroHaptics*, Vol. 3. 414–418.
- [45] Jonathan Lazar, Daniel Goldstein, and Anne Taylor. 2015. *Ensuring digital accessibility through process and policy*. Morgan kaufmann.
- [46] Cheuk Yin Phipson Lee, Zhuohao Zhang, Jaylin Herskovitz, JooYoung Seo, and Anhong Guo. 2022. CollabAlly: Accessible Collaboration Awareness in Document Editing. In *CHI Conference on Human Factors in Computing Systems*. Article 596, 17 pages.
- [47] Jaewook Lee, Jaylin Herskovitz, Yi-Hao Peng, and Anhong Guo. 2022. Image-Explorer: Multi-Layered Touch Exploration to Encourage Skepticism Towards Imperfect AI-Generated Image Captions. In *CHI Conference on Human Factors in Computing Systems*. Article 462, 15 pages.
- [48] Jiasheng Li, Zeyu Yan, Ebrima Haddy Jarjue, Ashrith Shetty, and Huaishu Peng. 2022. TangibleGrid: Tangible Web Layout Design for Blind Users. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. Article 47, 12 pages.
- [49] Joe Mullenbach, Craig Shultz, J Edward Colgate, and Anne Marie Piper. 2014. Exploring affective communication through variable-friction surface haptics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 3963–3972.
- [50] Joe Mullenbach, Craig Shultz, Anne Marie Piper, Michael Peshkin, and J Edward Colgate. 2013. Surface haptic interactions with a TPad tablet. In *Proceedings of the adjunct publication of the 26th annual ACM symposium on User interface software and technology*. 7–8.
- [51] Jakob Nielsen. 1994. *Usability engineering*. Morgan Kaufmann.
- [52] NVDA. 2020-10-15. Nv access. <https://www.nvaccess.org/>
- [53] Hari Prasath Palani, G Bernard Giudice, and Nicholas A Giudice. 2018. Haptic information access using touchscreen devices: design guidelines for accurate perception of angular magnitude and line orientation. In *International Conference on Universal Access in Human-Computer Interaction*. Springer, 243–255.
- [54] Yi-Hao Peng, Jeffrey P Bigham, and Amy Pavel. 2021. Slidecho: Flexible Non-Visual Exploration of Presentation Videos. In *The 23rd International ACM SIGACCESS Conference on Computers and Accessibility*. Article 24, 12 pages.
- [55] Yi-Hao Peng, JiWoong Jang, Jeffrey P Bigham, and Amy Pavel. 2021. Say It All: Feedback for Improving Non-Visual Presentation Accessibility. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. Article 276, 12 pages.
- [56] Beryl Plimmer, Andrew Crossan, Stephen A Brewster, and Rachel Blagojevic. 2008. Multimodal collaborative handwriting training for visually-impaired people. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 393–402.

- [57] Venkatesh Potluri, Liang He, Christine Chen, Jon E Froehlich, and Jennifer Mankoff. 2019. A Multi-Modal Approach for Blind and Visually Impaired Developers to Edit Webpage Designs. In *The 21st International ACM SIGACCESS Conference on Computers and Accessibility*. 612–614.
- [58] Venkatesh Potluri, John Thompson, James Devine, Bongshin Lee, Nora Morsi, Peli De Halleux, Steve Hodges, and Jennifer Mankoff. 2022. PSST: Enabling Blind or Visually Impaired Developers to Author Sonifications of Streaming Sensor Data. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. Article 46, 13 pages.
- [59] Nuzhah Gooda Sahib, Tony Stockman, Anastasios Tombros, and Oussama Metatla. 2013. Participatory design with blind users: a scenario-based approach. In *IFIP Conference on Human-Computer Interaction*. Springer, 685–701.
- [60] Anastasia Schaadhardt, Alexis Hiniker, and Jacob O Wobbrock. 2021. Understanding Blind Screen-Reader Users' Experiences of Digital Artboards. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. Article 270, 19 pages.
- [61] Carsten Schäfer. 2022. DrawerJS: a platform-independent, web-based JavaScript WYSIWYG HTML canvas editor. <https://www.drawerjs.com/>
- [62] Ather Sharif and Babak Forouraghi. 2018. evoGraphs — A jQuery plugin to create web accessible graphs. In *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. 1–4. <https://doi.org/10.1109/CCNC.2018.8319239> ISSN: 2331-9860.
- [63] Ather Sharif, Olivia H Wang, Alida T Muongchan, Katharina Reinecke, and Jacob O Wobbrock. 2022. VoxLens: Making Online Data Visualizations Accessible with an Interactive JavaScript Plug-In. In *CHI Conference on Human Factors in Computing Systems*. Article 478, 19 pages.
- [64] Ather Sharif, Andrew M Zhang, Anna Shih, Jacob O Wobbrock, and Katharina Reinecke. 2022. Understanding and improving information extraction from online geospatial data visualizations for screen-reader users (*Assets '22*). Athens, Greece, Article 61, 5 pages.
- [65] Mathieu Simonnet, Anke M Brock, Antonio Serpa, Bernard Oriola, and Christophe Jouffrais. 2019. Comparing interaction techniques to help blind people explore maps on small tactile devices. *Multimodal Technologies and Interaction* 3, 2 (2019), 27.
- [66] Jing Su, Alyssa Rosenzweig, Ashvin Goel, Eyal de Lara, and Khai N Truong. 2010. Timbremap: enabling the visually-impaired to use maps on touch-enabled devices. In *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services*. 17–26.
- [67] Shari Trewin. 1996. A study of input device manipulation difficulties. In *Proceedings of the second annual ACM conference on Assistive technologies*. 15–22.
- [68] Gregg C Vanderheiden. 1996. Use of audio-haptic interface techniques to allow nonvisual access to touchscreen appliances. In *Proceedings of the human factors and ergonomics society annual meeting*, Vol. 40. SAGE Publications Sage CA: Los Angeles, CA, 1266–1266.
- [69] Steven Wall and Stephen Brewster. 2006. Feeling what you hear: tactile feedback for navigation of audio graphs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06)*. Association for Computing Machinery, New York, NY, USA, 1123–1132. <https://doi.org/10.1145/1124772.1124941>
- [70] Lucy Lu Wang, Isabel Cachola, Jonathan Bragg, Evie Yu-Yen Cheng, Chelsea Haupt, Matt Latzke, Bailey Kuehl, Madeleine van Zuylen, Linda Wagner, and Daniel S Weld. 2021. Improving the accessibility of scientific documents: Current state, user needs, and a system solution to enhance scientific PDF accessibility for blind and low vision users. *arXiv preprint arXiv:2105.00076* (2021).
- [71] Jacob O Wobbrock, Krzysztof Z Gajos, Shaun K Kane, and Gregg C Vanderheiden. 2018. Ability-based design. *Commun. ACM* 61, 6 (2018), 62–71.
- [72] Jacob O Wobbrock, Shaun K Kane, Krzysztof Z Gajos, Susumu Harada, and Jon Froehlich. 2011. Ability-based design: Concept, principles and examples. *ACM Transactions on Accessible Computing (TACCESS)* 3, 3, Article 9 (2011), 27 pages.
- [73] Cheng Xu, Ali Israr, Ivan Poupyrev, Olivier Bau, and Chris Harrison. 2011. Tactile display for the visually impaired using TeslaTouch. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*. 317–322.
- [74] Tsubasa Yoshida, Kris M. Kitani, Hideki Koike, Serge Belongie, and Kevin Schlei. 2011. EdgeSonic: image feature sonification for the visually impaired. In *Proceedings of the 2nd Augmented Human International Conference (AH '11)*. Association for Computing Machinery, New York, NY, USA, 1–4. <https://doi.org/10.1145/1959826.1959837>
- [75] Yu Zhong, Walter S Lasecki, Erin Brady, and Jeffrey P Bigham. 2015. Regionspeak: Quick comprehensive spatial descriptions of complex images for blind users. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 2353–2362.