Empowering Agentic Non-Visual Web Navigation Through Tactile Controls and AI Support

by Amin Forootan

Submitted to OCAD University in partial fulfillment of the requirements for the degree of Master of Design in Inclusive Design

Toronto, Ontario, Canada, 2025

ABSTRACT

Blind and low-vision (BLV) users rely on screen readers to access digital content, yet these tools often impose strictly linear, text-based navigation that fails to communicate spatial layout, contextual changes, or emotional tone. This disconnect leads to cognitive overload, frustration, and reduced autonomy. In response, this research proposes a new model of screen reader interaction: a modular, tangible interface grounded in affordance-based design to enhance agentic non-visual navigation.

Through interviews and co-design sessions with BLV users, the study identifies six key experiential barriers: (1) loss of spatial orientation, (2) lack of state-change feedback, (3) absence of emotional/paralinguistic cues, (4) dependence on sequential logic, (5) inefficient input methods, and (6) mistrust of over-automated AI. These findings informed a series of design iterations, evolving from a conversational AI prototype to a tactile, multi-modal controller.

The final design features a rotary knob for sequential traversal, a rotor switch for hierarchical navigation, haptic and auditory feedback to signal changes and navigation boundaries, and a context-aware AI assistant. Mapped to NVDA screen reader commands and aligned with the POUR (Perceivable, Operable, Understandable, Robust) accessibility framework, each component reinforces spatial awareness, user agency, and reduced cognitive demand.

ACKNOWLEDGMENTS

Dr. Peter Coppin, Principal Advisor

Peter, thank you for creating the intellectual and creative space where this project could grow. Your guidance, curiosity, and belief in exploratory design processes helped shape both the direction and depth of this work. I look forward to many more tangents and insights together.

Dr. Alexis Morris, Secondary- Advisor

Alexis, thank you for your support, sharp insight throughout this journey. Your expertise in human-machine collaboration helped me think critically about the future of inclusive technology.

David Barter

David, thank you for your rigorous feedback, big-picture thinking, and generous conversations. Your ability to approach problems from different angles and your thoughtful feedback helped me refine conceptual aspects of this project. Participants

I also acknowledge the OCAD U School of Graduate Studies for providing Project & Travel Grant, essential funds toward prototyping, testing, and research participation support that was critical to the practical development of this research.

TABLE OF CONTENT

Creative Commons Copyright notice 2

Abstract 3

ACKNOWLEDGMENTS 4

Table of Content 5

List of tables & figures 7

1. Introduction 8

1.1 Background 8

1.2 Problem Statement 9

1.3 Research Questions 9

2. Literature Review 9

3. Methodology 11

3.1 Participants 12

3.2 Overview of Research Activities 12

3.3 Semi-Structured Interviews 12

3.4 Co-Design Series 13

3.5 Prototype Development 13

3.5.1 Iteration 1: Conversational AI Assistant 13

3.5.2 Iteration 2: Rotary Encoder controller for NVDA Screen Reader Users. 17

3.5.3 Iteration 3: Modular Controller with Knob + Rotor Switch + AI + Vibration 19

3.6 Usability Testing, Data Collection Analysis 22

4. Findings 23

4.1 Inductive Insights from Initial Interviews 24

4.2 Iteration 1: Conversational AI Assistant Prototype 26

4.3 Iteration 2: Rotary-Based Screen Reader Controller 29

4.4 Iteration 3: Modular Controller with Joystick 33

4.5 Final Design Suggestion 38

5. Discussion 42

5.1 Accessible Content & Semantic Structure 42

5.2 Layout Complexity & Navigation Map 43

5.3 Input Modalities & Interaction Methods 43

5.4 Output Modalities & Feedback Mechanisms 43
5.5 Customization & Personalization Settings 43
5.6 User Skills, Experience, and Strategies 43
5.7 Summary 44
6. Conclusion 47
7. References 48
8. APPENDICES 51
8.1 appendix A- Arduino Code Iteration 2 51
8.2 appendix A- Python Code Iteration 2...54

LIST OF TABLES & FIGURES

Table 1: components used in iteration 2 18

Table 2: components used in iteration 3 20

Table 3: available interactions in second prototype 29

Table 4: available interactions in third prototype 34

Table 5: affordance-based evaluation across UX dimensions 43

Figure 1: chatbot interface in Voiceflow 16

Figure 2: Voiceflow node-based interaction design 17

Figure 3: hardware setup of iteration 2 19

Figure 4: hardware setup of iteration 3 20

Figure 5: design spiral of the controller project 23

Figure 6: Wizard-of-Oz AI assistant test 27

Figure 7: rotary knob testing with participant 30

Figure 8: schematic of knob-based navigation model 31

Figure 9: failed goal reach with knob-only navigation 32

Figure 10: third prototype with knob and joystick 33

Figure 11: schematic of dual-input screen reader interface 36

Figure 12: inefficient interaction path using joystick and knob 37

Figure 13: annotated controller layout 38

Figure 14: rotor switch enabling semantic layer control 39

Figure 15: layered navigation with rotor and knob 40

Figure 16: ergonomic design of final controller form factor 41

1. INTRODUCTION

1.1 Background

Screen reader technologies provide essential access to digital content for blind and low-vision (BLV) users. Yet despite their utility, these interfaces remain constrained by linear, text-based approaches that often obscure spatial structures, suppress paralinguistic cues, and hinder the provision of context for users' comprehension. As a result, BLV users must traverse complex digital environments with limited non-spatial feedback, leading to slower performance, cognitive overload, and diminished autonomy. These limitations highlight a critical gap in the design of inclusive, non-visual interaction models.

In response, this project presents a modular, tangible controller that reimagines screen reader interaction through affordance-based design. Affordances, as described by Gaver (1991), are properties of the environment that align with users' abilities to act. When affordances are perceptible, they create an intuitive link between perception and action, allowing users to understand how to interact with technology directly through its physical or sensory characteristics. In this context, the controller is designed to make navigation actions more perceivable and actionable through tactile and auditory feedback rather than relying on combinations of keyboard keys stored in limited working memory and symbolic or abstract representations. The system integrates a rotary knob for sequential traversal, a rotor switch for layered navigation, and embedded auditory, haptic, and AI-assisted feedback. This controller supports a more intuitive, expressive, and user-directed experience by enabling users to bypass strict linearity and access content using fast navigation and layer switching.

This work combines user-centered methods and iterative design, beginning with a review of literature on non-visual access and interface cognition. It proceeds through co-design sessions, three design iterations, and feedback from BLV participants. These iterations, from a conversational AI prototype to a refined physical interface, are guided by six recurring experiential challenges identified in early interviews. The final chapters outline the resulting design framework, theoretical contributions, and next steps for validation.

By shifting away from command-based, linear navigation toward multi-modal, user-initiated control, this research contributes both a novel non-visual interaction interface and a transferable conceptual model for enhancing non-visual digital interaction.

1.2 Problem Statement

Current screen reader systems do not adequately support non-visual skimming, layered control, or spatial orientation. BLV users are forced to rely on rigid, memory-intensive navigation strategies that hinder their ability to interact fluently with complex digital environments.

1.3 Research Questions

- What are key affordances that support more agentic screen reader use and how to optimize them?
- How can AI be integrated as a supportive assistant without undermining user agency?

2. LITERATURE REVIEW

Screen Reader technologies have opened many doors, but also impose distinct cognitive, spatial, and emotional burdens. These limitations, documented in both academic and applied research, such as the Accessible Standards Canada (ASC) report on strategies for improved inclusive virtual ICTs (Coppin, Hung, & Uribe Quevedo, 2024)), do not stem purely from the tools themselves, but from a mismatch between screen reader logic and interface design practices that often neglect non-visual users.

This research critiques the dominant model of linear, sequential reading by focusing on gaps in spatial understanding and contextual awareness, an alignment with the experience of "not knowing what you don't know" (Bigham et al., 2017). The ASC report (Coppin et al., 2024) similarly emphasizes the need for more intuitive, modular alternatives to traditional screen reader interaction and recommends exploring what it refers to as a "controller paradigm" for improving navigation and autonomy.

Navigating a complex web interface through a screen reader is rarely intuitive; users must rely on memory, repetition, and patience to complete even simple tasks. This mismatch is not solely due to the limitations of screen readers themselves but is equally driven by sighted designers deprioritizing non-visual access in favor of visual-centric layouts.

Larkin and Simon's (1987) theory of sentential versus diagrammatic representations frames this tension well: while sighted users benefit from spatial layouts that group and relate information visually, screen readers flatten everything into a serial stream. The cost of search, in this case, the cognitive effort, time, and navigation complexity required to locate relevant information, as Larkin and Simon (1987) describe, rises sharply, and users must endure rigid, memory-intensive workflows that obscure structure, relationships, and meaning.

Studies over the past two decades have repeatedly identified these tensions. Lazar et al. (2007) and Murphy et al. (2008) documented recurring pain points, including inaccessible layouts, poorly labeled elements, and form designs that confuse rather than guide. More recently, Williams et al. (2019) emphasized how linear navigation often disrupts user autonomy and increases the risk of missing critical content.

Yet the issue is not just functional, it's experiential. Lagman (2016) highlights the emotional and cognitive toll that current screen reader tools can impose. Bigham et al. (2017) coined the phrase "not knowing what you don't know" to describe a common experience of BLV users: the inability to detect omitted information, broken flows, or hidden structure. These insights have prompted a shift in accessibility research from compliance to lived experience, from making content merely available, to making it meaningfully usable.

To address these gaps, several researchers have explored strategies to reduce linearity and reintroduce spatial orientation. Ahmad (2012) proposed skimming methods that allow users to scan without reading sequentially. Dissanayake (2015) developed a browser tool that previews web page structure, such as menus and headings, before engagement, supporting mental mapping and reducing trial-and-error. Coppin et al. (2024) introduced non-linguistic guided tours to convey structure through cues such as rhythm, tone, or spatialized sound. These interventions recognize that structure matters not just for efficiency, but for confidence and orientation.

A growing body of work also explores the use of conversational AI to supplement or reframe interaction. Baez et al. (2022) proposed a natural language framework for web browsing, allowing users to ask for summaries, checkpoints, or context-sensitive help. Zhang et al. (2023) developed "Creator," a conversational screen reader for blind content creators, incorporating features such as guided narration and AI-scaffolded control. These tools shift interaction from reactive to proactive, allowing users to steer the experience rather than passively receive it. However, user studies also caution against over-automation. As Hegde (2023) notes, users prefer AI that assists, but does not override, their strategies.

Parallel to these developments, researchers have increasingly explored multimodal and affordance-based approaches to screen reader interaction.

Emotional engagement has likewise emerged as a design concern. Bragg et al. (2018) observed that although BLV users often process synthetic speech faster than sighted peers, satisfaction depends on voice type, delivery, and tone. Choi et al. (2020) found that slower, more humanlike speech fostered emotional intimacy and reduced fatigue. Graham's (2015) Accessible User Experience (AUX) framework builds on this, advocating for metrics such as emotional comfort, user agency, and affective clarity, principles that underpin the AI assistant in this project.

Customization remains another critical factor. Jordan et al. (2024) proposed five personas representing different screen reader strategies, emphasizing that one-size-fits-all models are inadequate. Borodin et al. (2010) similarly noted how BLV users often invent personalized workarounds to navigate inaccessible content, highlighting the importance of modular, user-driven design. These findings validate the inclusion of tactile mode-switching and on-demand AI in this project's final prototype.

Finally, multimodal paradigms like VERSE (Vtyurina et al., 2019) point to promising hybrid interfaces. Combining screen reader controls with voice, gestures, and spatial audio, VERSE allowed users to navigate with greater fluidity and lower command overhead. Fink et al. (2024) applied similar multimodal principles to vehicles, using tactile and auditory cues to deliver layered environmental data. These systems prioritize user control, perceptual clarity, and contextual awareness, all of which shaped this study's final interaction model.

In summary, the field is moving toward more human-centered, multimodal, and emotionally responsive tools for non-visual access. Yet the need remains for tactile, spatial interfaces that respect user strategies, restore orientation, and offer optional support without taking over. This project contributes to that conversation by integrating AI-driven layers into complex digital navigation in a more intuitive and empowering experience, an effort inspired in part by recommendations from the ASC report to develop controller paradigms that address the limitations of traditional screen readers.

3. METHODOLOGY

This section outlines the methodology used in designing and evaluating a novel screen reader controller for blind and low-vision (BLV) users. Grounded in human-centered design and the principles of inclusive research, the process combined interviews, co-design, iterative prototyping, and usability testing to explore how tangible interaction and AI assistance can improve screen reader navigation.

3.1 Participants

Participants were recruited through targeted outreach to accessibility networks, including mailing lists of assistive technology users, word-of-mouth, and personal referrals Perceptual Artifacts Lab, OCAD University. The goal was to involve individuals who regularly use screen readers and have lived experience navigating digital interfaces non-visually. All participants provided informed consent.

A total of five BLV participants took part in this study, representing a range of screen reader experiences, from daily NVDA (NonVisual Desktop Access tool) and VoiceOver (Apple screen reader) users to those who occasionally relied on assistive tech. Participants varied in age (from mid-20s to mid-60s), digital proficiency, and navigation preferences. Some were highly experienced with keyboard shortcuts and VoiceOver gestures, while others favored slower-paced or tactile interfaces.

3.2 Overview of Research Activities

The research proceeded in five main stages:

- Semi-Structured Interviews to gather experiential insights and challenges with screen readers.
- Participatory Co-Design Sessions, spread across multiple sessions, to iteratively develop and refine interface concepts.
- Wizard-of-Oz Testing of AI Assistance, simulating contextual help during web browsing.
- Physical Prototyping with Arduino, culminating in three working prototypes.
- Functional Usability Testing of the tactile controller and AI integration with real-world browsing tasks.

3.3 Semi-Structured Interviews

Initial interviews were conducted using a conversational format. Each session lasted 60–90 minutes and explored user frustrations, strategies, and ideal navigation scenarios. Participants were asked about:

- Daily navigation habits (keyboard vs. touch).
- Emotional experiences of autonomy or frustration.
- Expectations for AI and voice-based support.
- Preferences for feedback mechanisms (audio, tactile, etc.).

Transcripts were reviewed to identify common barriers, such as spatial disorientation, redundant speech output, and low trust in voice agents. Think-aloud methods were employed during browsing tasks to externalize user thought processes.

3.4 Co-Design Series

Four iterative sessions were conducted with key participants. Each session focused on a different input modality or feedback system:

- Session 1: AI voice assistant check, a Wizard-of-Oz (Bernsen et al., 1994) simulation to evaluate contextual help.
- Session 2: Mid-fidelity prototypes using a rotary encoder were tested for intuitiveness in directional control.
- Session 3: Mid-fidelity prototypes using a rotary encoder and joysitck with AI feature integration.

3.5 Prototype Development

Prototypes were built across three iterations:

3.5.1 Iteration 1: Conversational AI Assistant

A chatbot was developed using the Voiceflow platform. Voiceflow is a collaborative, no-code tool that enables teams to design, prototype, and deploy conversational AI agents—such as chatbots and voice assistants, across multiple channels through a visual drag-and-drop interface. It supports integration with major AI models and APIs, allowing for the creation of sophisticated, multi-channel conversational experiences without requiring programming expertise. In this project, the chatbot simulated screen-reading assistance by providing emotionally resonant, non-visual descriptions, though it lacked interruption control. A total of 44 screenshots were taken from the Northern Reflections online shop, which was selected based on participant feedback during interviews—participants identified it as an accessible webpage they regularly use to purchase clothing. Descriptions for the screenshots were generated using ChatGPT.

We developed a conversational experience with an AI assistant tailored to blind users, focused on navigating a digital retail interface. The activity was designed as a co-exploration of inclusive e-commerce accessibility, emphasizing rich, descriptive, non-visual communication. Our process involved uploading sequential screenshots of an online clothing store and prompting the AI to generate concise, useful, and emotionally resonant descriptions that blind users could use to understand and navigate the interface.

The goal was to simulate the experience of an AI assistant guiding a blind user through an online shopping interface using image descriptions and conversational decision-making. We began by uploading the series of screenshots from the Northern Reflections website. These images included:

- The landing page and homepage hero banners.
- Category menus and filters.
- Product carousels.
- Individual product pages with details, options, and pricing.
- Customer reviews.

To achieve clarity and usability for blind users, we established a consistent format for all responses. Prompts were crafted carefully to guide the AI in producing:

- Clear, emotionally descriptive explanations of images.
- Options organized by UX/UI priority (following signal detection theory).
- A conversational tone as if the webpage is "speaking" to the user.
- No spatial or visual-only references (e.g., no "on the right" or "in the corner").
- Language designed to convey function and feeling over layout or appearance.

An early example of a successful prompt format that attached to Chat-GPT 4o was:

"Assume that you are a friend of a blind person and you want to explain this page to him. Give the options available in terms of clickable links and buttons, sorted based on the level of importance that you catch from the UX/UI design perspective. Explain the image and text. Try to use words to convey the feeling. Keep it as short as possible. Do not use spatial description or abstract language."

This structure was used repeatedly with minor adjustments, resulting in a consistent interaction model that mimicked a live assistant.

The AI (Chat-GPT 4o) provided a concise and mood-driven overview of the main page, including promotional banners. It described messages like "Timeless Style, New Looks" alongside brief descriptions of models such as "confident blond woman," followed by prioritized options for navigation: viewing the menu, checking promotions, using search, or scrolling down.

Also, We uploaded screenshot of the site's menu structure. The assistant presented each category (e.g., "New Arrivals," "Apparel," "Deals & Sale") as selectable options, with a final user-facing prompt asking:

"Would you like to select an option or return to the last page?"

Each product was described with a consistent structure including:

- Price and rating summary.
- Item type (e.g., "Sleeveless blouse").
- Occasion and fabric feel (e.g., "ideal for casual warm days," "soft and breathable").
- Cut, length, and key features (e.g., "waist-length," "ruffle details").
- Colors conveyed with emotional associations (e.g., "blue like the feeling of calm water").
- A two-word persona description of the model (e.g., "cheerful and calm").

After complete explanation of each item Users were asked:

"Would you like to check the next item, or go back to the main menu, check promotions, or use the search bar?"

Filter categories (e.g., color, fit, product type, and price range) were broken down into conversational options:

"Do you want to filter by color, fit, type, or price? I can read the options or you can cancel."

After applying search or filters, users were presented with a summary:

"Search results are ready. You can sort or filter further, or explore the items one by one. Each product includes price, reviews, star rating, and color options."

- For each product page, the chat bot described:
- Product name, price, available payment plans
- Available color swatches
- Fit (e.g., Regular or Petite)
- Sizes from XS to XL

All the descriptions created during this activity were integrated into a working flow using the Voiceflow tool. We used these written outputs to build a voice assistant chatbot tailored for blind users. The tone of the assistant was adjusted in Voiceflow to maintain consistency with the descriptive, supportive, and choice-oriented language style developed through the prompts.

During live sessions with blind participants, we used a Wizard of Oz method. As the AI-generated descriptions and available options became clear, we acted on behalf of the system, selecting the appropriate next steps based on the participant's spoken response. This allowed us to simulate a dynamic voice assistant experience, where the chatbot appeared to respond directly to the user's choices in real time. This method helped validate the structure and flow of the voice interaction before full automation.

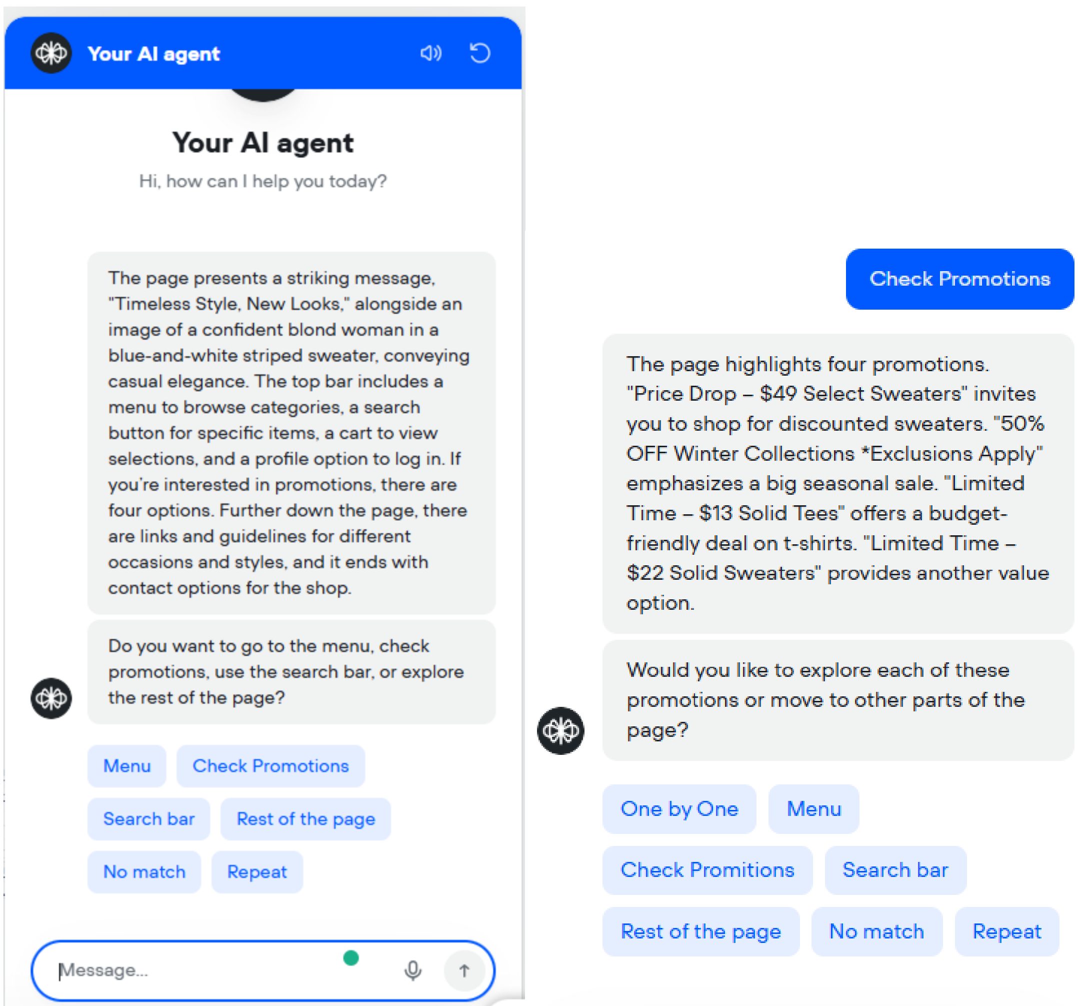


Figure 1. *Voiceflow chatbot interface simulating screen reader interaction. The left image shows the initial description of a webpage, highlighting its layout, visual content, and navigation options such as the menu, search bar, and promotional sections. The right image presents a deeper exploration of the "Check Promotions" section, listing four specific promotional offers and prompting the user to continue navigating through labeled options.*

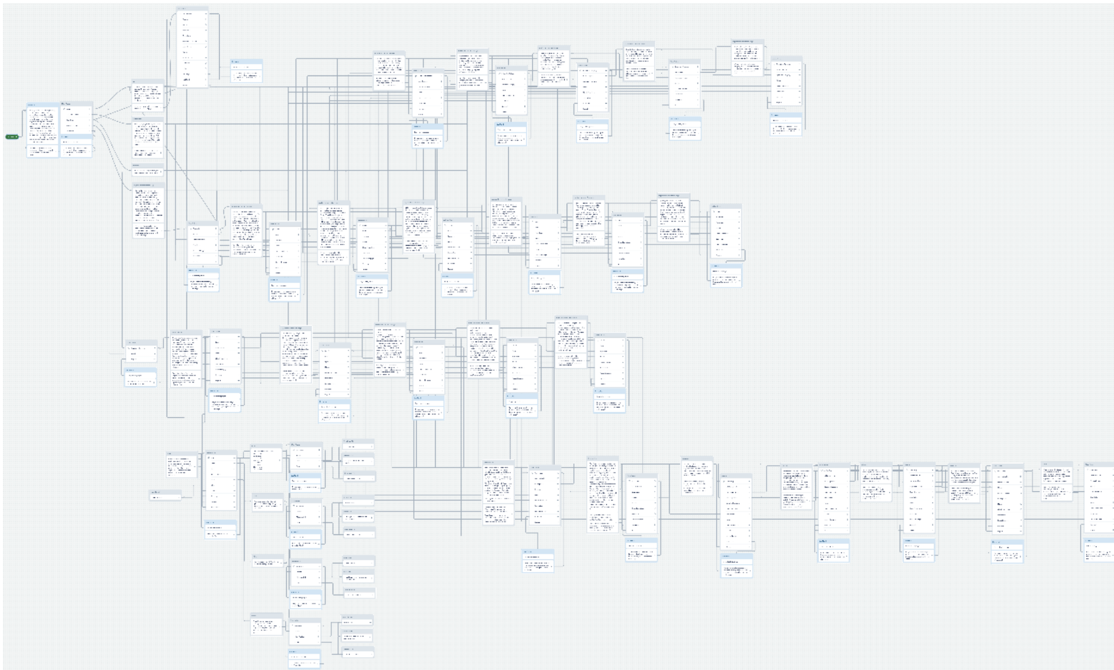


Figure 2. *Voiceflow interface showcasing the full node-based design of the chatbot interaction. The diagram visualizes how user responses guide the flow between connected conversational blocks, simulating screen reader navigation across a structured webpage.*

3.5.2 Iteration 2: Rotary Encoder controller for NVDA Screen Reader Users.

A tactile controller mapped rotary movements to NVDA navigation keys. Feedback focused on the intuitive metaphor of scanning, but raised concerns about scroll speed and lack of layered control. This tool provides a tangible alternative input method for blind and low-vision (BLV) users navigating digital interfaces using the NVDA screen reader. It leverages a rotary encoder with an integrated push button to trigger essential NVDA commands, including forward and backward focus traversal, heading navigation, and activation of elements. The system is optimized for low-cost implementation and real-time responsiveness and is intended to augment or partially replace traditional keyboard navigation through a single-hand, tactile controller.

Table 1. *Components used in iteration 2*

Component	Specification
Microcontroller	Arduino Nano 33 IoT
Rotary Encoder Module	Mechanical encoder with push-button switch
Jumper Wires + Breadboard	Standard prototyping setup
Host Computer	Windows 10/11 with NVDA screen reader installed
Python Environment	Python 3.10+ with pyserial and keyboard libraries

The Arduino code reads rotary encoder movement and button presses. It transmits serial commands to the host machine via USB serial Rotary position is not bounded artificially, allowing continuous interaction in both directions without lockout at a lower limit. The codes are available in the appendix A.

A Python listener script runs on the host machine. It Connects to the correct COM port (communication channel ID to the laptop) at 9600 baud (transmission speed), Listens for incoming serial strings ("tab", "shift_tab", "H") and Translates them into OS-level key presses using the keyboard library. Implements double-click detection via timestamp comparison to map two rapid "H" presses to Enter. Double-clicks are identified within a 300ms threshold using time.time() deltas. Single taps produce the H keypress immediately unless followed by a second tap within the threshold. The code is available in the appendix B. Debounce logic ensures that both single and double clicks are responsive while avoiding unintentional triggering. Minimal training is required. Users familiar with NVDA will find the mapped keys intuitive and standard.

This controller provides an accessible and intuitive interface for NVDA screen reader users using minimal hardware. It leverages core navigation shortcuts via tactile gestures and introduces a compact, one-handed input alternative to full keyboard use. The system is suitable for research, usability testing, and development toward assistive navigation products.

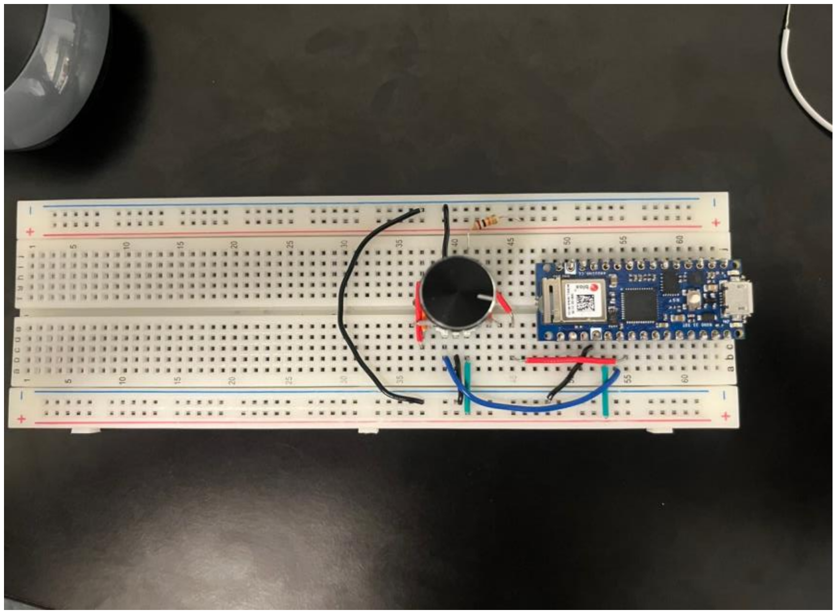


Figure 3. *Hardware setup for the second iteration of the screen reader controller. The Arduino Nano 33 IoT and a rotary encoder with push-button functionality are connected on a breadboard using standard jumper wires. The device transmits serial commands (e.g., "tab", "shift_tab", "H") to a Python listener script running on a Windows host machine with NVDA installed.*

3.5.3 Iteration 3: Modular Controller with Knob + Rotor Switch + AI + Vibration

The third iteration of the NVDA screen reader controller builds on the initial tactile navigation design by integrating a modular hardware setup with enhanced input modes and AI-based contextual assistance. Designed for blind and low-vision (BLV) users, this version introduces a joystick, expanded rotary encoder functionality, and GPT-4 Vision support via NVDA's AI Content Describer add-on. The system aims to support intuitive, plug-and-play controller designs. This iteration was guided by three main goals:

- Enable multidirectional navigation via joystick for improved control granularity.
- Introduce AI-supported interface understanding through image-based description tools.
- Ensure full plug-and-play functionality using HID-enabled microcontrollers with no external Python runtime requirements.

Table 2. *Components used in iteration 3*

Component	Specification
Microcontroller	Arduino Pro Micro (Leonardo-compatible)
Rotary Encoder Module	Mechanical encoder with push-button switch
Joystick Module	2-axis analog joystick with digital push button
Wiring	Standard jumper wires and breadboard
Host Machine	Windows 11 PC running NVDA + AI Content Describer add-on

Joystick directional input is interpreted using analog readings from the VRx and VRy pins on Arduino. Modifier mode is toggled based on button state and affects encoder mappings.

To support real-time contextual understanding of interface layouts and images, this iteration integrates NVDA's AI Content Describer add-on. This plugin allows users to send a screenshot to a large vision-language models such as GPT-4 Vision and receive a detailed spoken description.

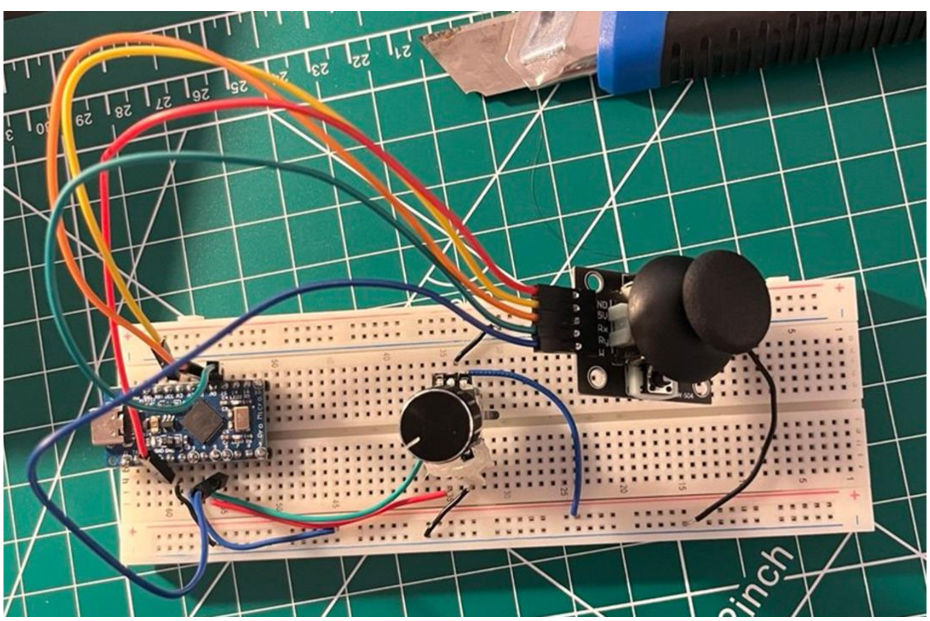


Figure 4. *Hardware setup for the third iteration of the screen reader controller. An Arduino Pro Micro, rotary encoder, and 2-axis joystick are connected on a breadboard using standard jumper wires. Joystick input and encoder gestures are interpreted by Arduino and transmitted to running NVDA with the AI Content Describer add-on.*

When a user presses and holds the rotary knob for over one second, the controller triggers NVDA's screenshot shortcut. NVDA captures the current screen, submits it to the AI model via API, and reads the returned description aloud. This replicates functionality found in tools such as Be My Eyes (Be My Eyes is a free app that connects blind and low-vision users with sighted volunteers or AI assistants for visual support.) AI, but integrates seamlessly into the NVDA workflow. The benefits of the approach are:

- Integrates with any visual web interface without switching apps.
- Reduces ambiguity by targeting the screen content that the user is actively navigating.
- Extensible with additional prompt control and image cropping in future iterations.

To improve the relevance and accessibility of AI-generated responses, a refined system prompt is used:

"Describe this screen to a blind user. Focus on the red-highlighted section of the image, which represents the current area being navigated by screen reader. Avoid spatial references and emphasize headings, button labels, product names, and visual cues."

This ensures that AI outputs reflect screen reader behavior, avoid inaccessible descriptions like "on the left," and enhance usability for BLV users.

A lightweight Python script was used for internal testing to capture and save screenshots on an AI trigger. This code annotates the screenshot with a visible red rectangle (mimicking NVDA's virtual cursor or highlighted element). This enables visual debugging and cross-checks with AI output for research and gives more content to the AI to describe inside the red box which makes the tool context aware. This process can be fully automated in future iterations using NVDA's internal focus object APIs or by integrating mouse routing to the navigator object.

NVDA's Focus Highlight feature was tested, though not programmatically manipulated. This guided how red-box annotations could assist BLV testers. Alternatives such as LLaVA (Large Language and Vision Assistant) and Be My Eyes were tested. GPT-4o Vision proved the most context-aware and accurate for describing screenshots.

3.6 Usability Testing, Data Collection Analysis

Participants tested the prototype across tasks such as:

- Finding product information on an e-commerce page
- Navigating a visual-heavy news website
- Accessing a help page and locating contact information

Feedback was recorded and annotated. In some cases, a second observer noted gesture patterns and verbal cues of hesitation or delight. Data was collected through recorded interviews, observation notes, and direct participant quotes. Analysis followed a qualitative thematic approach:

- Open Coding of quotes and behavior
- Affordance Mapping of device features to experiential needs
- Cross-User Synthesis to identify patterns across participants

Triangulation across interview themes, prototype use, and follow-up feedback helped validate insights. Participant anonymity was maintained through pseudonyms and redacted transcripts.

4. FINDINGS

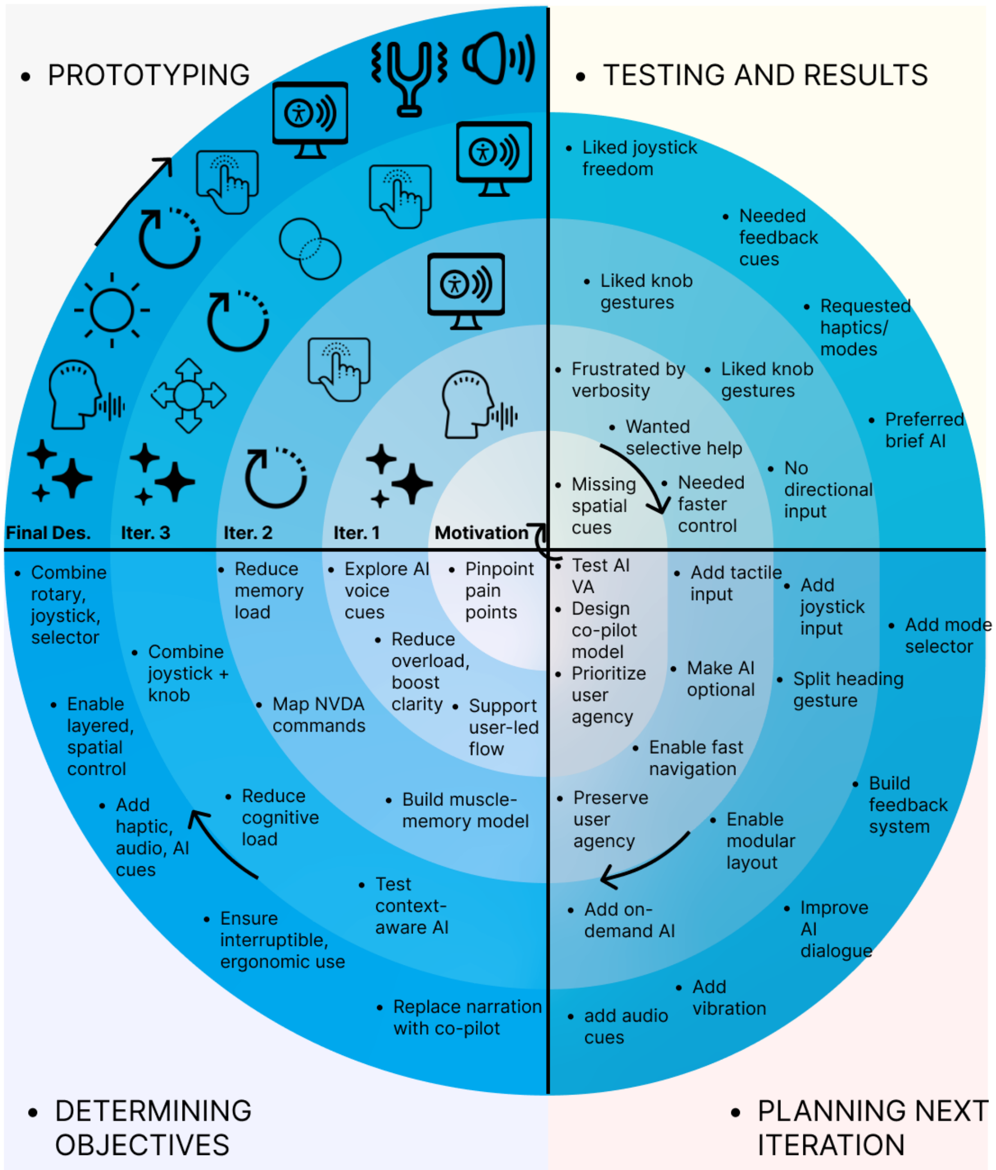


Figure 5. *Design Spiral of the Screen Reader Controller Project.*

The infographic in Figure 5 visualizes the full design process of our modular screen reader controller for blind and low-vision users. This visualization was inspired by the participatory and iterative design approach demonstrated in the development of a 3D audio-tactile globe for non-visual users (Ghodke, Yusim, & Somanath, 2019). Starting with open-ended interviews and theoretical framing (center), we identified cognitive, emotional, and spatial challenges with existing screen readers. Guided by user feedback, we iteratively developed three prototypes: a conversational AI assistant (Iter. 1), a rotary-based tactile controller (Iter. 2), and a joystick-enhanced modular device (Iter. 3). Each iteration followed a spiral process of defining objectives, prototyping, testing, and planning next steps. Participatory design and co-design sessions informed every stage. The outer ring represents the final design proposal, which integrates tactile input, mode change capability, and context-aware AI support.

4.1 Inductive Insights from Initial Interviews

To begin our design process, we chose to listen first. We spoke with blind and low-vision (BLV) users about how they navigate digital environments, not only what works or doesn't, but how it feels, how they adapt, and what they wish they could do differently. Rather than entering with a fixed plan, we started with open-ended questions: How do BLV users manage the absence of visual structure? What are the cognitive and emotional burdens imposed by current screen reader tools? Where are they forced to compensate, and how?

As these stories unfolded, we began recognizing consistent tensions between user needs and interface behaviors. To help interpret them, we turned to the cognitive framework developed by Larkin and Simon (1987), who differentiate between sentential and diagrammatic representations. Sentential representations, such as the output of a screen reader, are linear and sequential. They deliver one piece of information at a time, requiring the user to store, compare, and synthesize content in memory. Diagrammatic representations, by contrast, use spatial layout to communicate meaning through clustering, alignment, and hierarchy, allowing users to visually recognize patterns, infer relationships, and make decisions more efficiently.

The experiences described by our participants echoed many of these conceptual mismatches. One user described trying to navigate a shopping website: "*Amazon's home page is a nightmare. Tons of promotions and irrelevant information before you get to anything useful. I skip all that and just search directly.*" This struggle reflected the concept of search cost, defined by Larkin and Simon as the mental effort and time required to locate relevant information when navigating sententially (1987, p. 69). The ASC report (Coppin et al., 2024) reinforces this issue, identifying high "search cost" as a key limitation of linear audio navigation and recommending 2D navigation paradigms such as grid-based interfaces or physical controllers to reduce unnecessary traversal. In a diagrammatic layout, irrelevant content can be skimmed or ignored. In contrast, screen readers enforce a linear path that makes all content equally weighted and time-consuming to parse. The ASC report (Coppin et al., 2024) also highlights the need for spatial models, noting that traditional screen reader navigation "removes the GUI's spatial affordances," and suggests solutions such as mini-indexes and neighborhood scanning to regain orientation. From this, we began asking: how can we reduce this search cost by curating or filtering what is read based on the user's intent?

Another recurring challenge was the loss of spatial context. Larkin and Simon (1987) refer to this as the absence of spatial indexing. One participant put it simply: "*I can't just jump between what matters. I have to go through every item unless there's a heading. And sometimes, there isn't one.*" Without spatial anchors such as visual groupings or layout conventions, users must rely entirely on memory or semantic cues to maintain orientation. They instead depend on inconsistent features such as headings, which act as improvised landmarks. This led us to ask whether auditory or semantic structuring could simulate the experience of spatial indexing and help users form a more intuitive mental map of the interface.

This loss of structure also forced users to rely more on inference than recognition. Recognition, as defined by Larkin and Simon (1987), involves identifying meaning directly from presentation, such as recognizing that bold text signifies a title or that grouped items belong together. Sentential systems require inference, meaning that users must deduce relationships by integrating separate pieces of information over time. Our participants described having to reconstruct the page layout mentally based on what they had heard a few moments earlier. "*Headings are like signs on a highway. Without them, I'd be driving blind,*" one participant said. These insights brought up a new challenge: how might we design for recognition in non-visual formats, easing the cognitive strain caused by constant inference?

Participants also spoke about the loss of paralinguistic and affective cues. "*It just says 'star emoji,' but I don't know what that means without context,*" one user noted. Another reflected, "*If my screen reader doesn't tell me something's important, it's gone.*" These comments point to the absence of emotional and contextual clarity, which visual interfaces typically convey through boldness, spacing, icons, or color. These expressive layers are flattened in a sentential stream. Bragg et al. (2018) and Choi et al. (2020) both found that voice type, pacing, and tone impact comprehension and emotional engagement. The ASC report similarly noted that auditory saturation without affective modulation leads to sensory fatigue, advocating for multimodal feedback. From this, we asked: could voice tone, adaptive summaries, or AI-supported phrasing restore some of the expressive functions lost in the transition to audio?

Just as important as content delivery was the issue of control. "*Too much detail is exhausting. I just want to ask for what I need*," one participant said. Another shared, "*I don't need everything. I want to ask what I need, not wait for the whole thing.*" These comments emphasized the limitations of rigid, linear systems that lack iterative refinement, a flexibility often available in visual interfaces. In screen readers, users are often locked into a one-way stream. The ASC report (Coppin et al., 2024) recommends interaction paradigms that allow users to control granularity, jumping across categories, skipping cells, or toggling verbosity. This insight sparked a design question: could a conversational system support back-and-forth inquiry, allowing users to refine their interaction dynamically?

Finally, many participants emphasized that not all tasks require the same kind of assistance. For form-filling or login tasks, step-by-step guidance was seen as useful. For browsing or reading reviews, participants preferred summaries or overviews. A one-size-fits-all model would be inadequate in too many regards. This highlighted a final question for design: how can we create a system that adapts its level of detail and support based on the task and the user's needs in the moment?

Together, these reflections led us to a clearer understanding of the cognitive constraints imposed by screen readers and the types of affordances they eliminate. Our participants confirmed that when information is stripped of spatial organization, paralinguistic expression, and navigational flexibility, users are left with a cognitively demanding and emotionally flattened interaction model. The result is fatigue, inefficiency, and a reduced sense of control.

By anchoring each insight in both user feedback and cognitive theory, we arrived at a set of design questions that would guide our first iteration. Could a conversational assistant reduce search cost by summarizing and highlighting relevant information? Could it reintroduce the expressiveness and spatial logic of visual interfaces through tone, structure, and feedback? Most importantly, could it do so without overriding the user's strategies, giving them agency instead of automation?

These questions did not prescribe a solution. They opened a direction to design not just a tool, but a co-pilot, something that listens, responds, and supports, without ever taking control.

4.2 Iteration 1: Conversational AI Assistant Prototype

The first design iteration explored whether a conversational AI assistant could support blind users by improving emotional tone, navigational clarity, and interaction fluidity in online tasks. Built using the Voiceflow platform (Voiceflow is an open-source AI tool to design AI chatbots), this prototype aimed not to replace screen readers but to reimagine their underlying interaction logic through natural language, task-specific support, and emotionally resonant delivery. The testing scenario involved browsing an online clothing store, navigating categories, reviewing products, and performing actions such as "add to cart."

The assistant provided emotionally attuned prompts, relying on sensory metaphors (e.g., warmth, texture) rather than visual references. It summarized reviews by sentiment, offered save-for-later functionality, and provided product overviews in a friendly voice. To simulate intelligent interaction, a Wizard of Oz approach (Bernsen et al., 1994) was used: the researcher manually selected appropriate responses based on participants' verbal cues (see Figure 6).

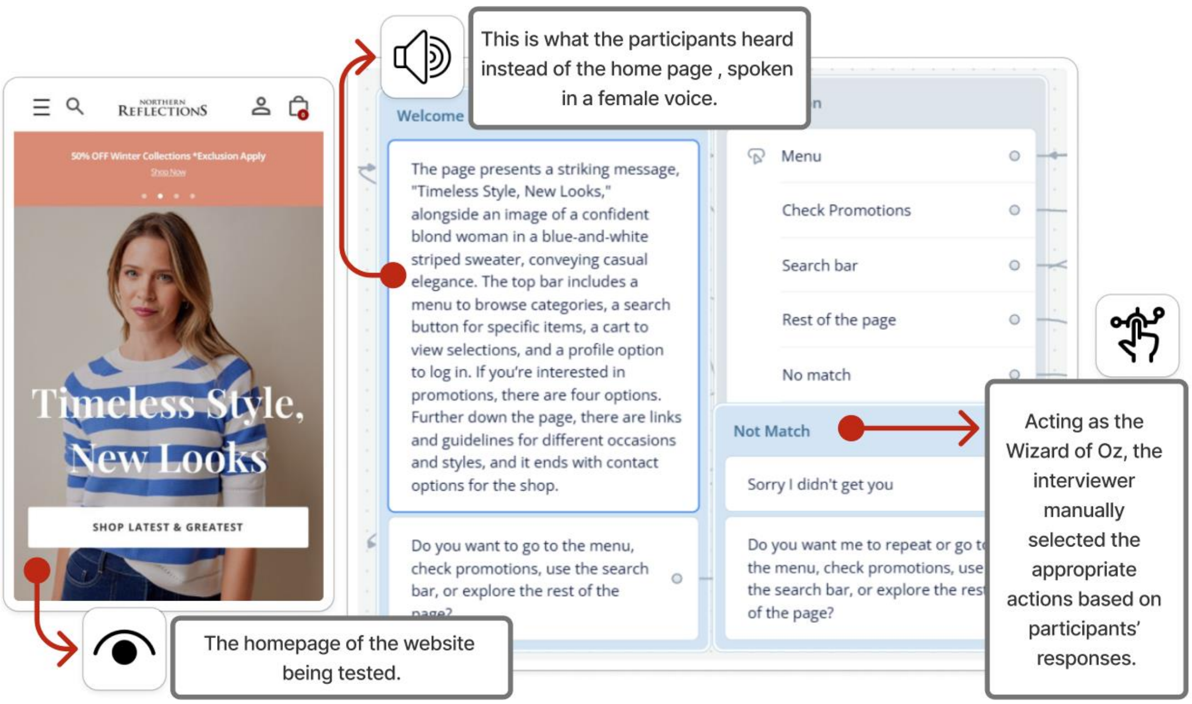


Figure 6. *Simulated AI voice assistant interaction: The fashion homepage was translated into spoken descriptions, with an interviewer manually responding to participant commands, mimicking intelligent voice-based navigation.*

Participants expressed appreciation for the assistant's voice tone and the structured way information was categorized. Yet this appreciation was tempered by frustration with the lack of control. "*Too much detail is overwhelming*," one participant noted, especially when the assistant could not be interrupted mid-sentence. Another reflected, "*It was too polite to be useful*," articulating the desire for faster, more responsive dialogue.

A key theme that emerged was the importance of maintaining user agency. Participants emphasized that the assistant should act as a co-pilot, not a narrator. As one put it: "*I want screen readers and voice assistants to give me clear options without overloading me.*" They rejected the idea of an assistant that guided them through every interaction. Instead, they wanted something more modular, interruptible, and responsive, something closer to real-life help, where they might only ask someone sighted for confirmation or support at specific moments. This insight parallels findings in the ASC report, which emphasized that assistive systems should "support selective engagement" rather than enforce full control or passive consumption (Coppin et al., 2024, Section F.4.5).

Another insight concerned how BLV users approach browsing. While sighted users often engage in exploratory or casual browsing, participants explained that BLV users tend to begin with a focused intent. "*I want to start broad, like a search for coats, and then filter quickly.*" The prototype's default behavior, slow, linear delivery of all product details, directly undermined this task-driven strategy. This mirrors findings in the ASC report, which noted that constraint-driven user behavior in BLV participants often stems from inaccessible exploratory modes, not from preference.

Trust and reliability were also major factors. While participants appreciated the assistant's tone, they remained skeptical of AI fully handling the task. *"I feel like I'm missing something,"* one participant admitted. This reflects a broader pattern observed across accessibility research: BLV users want AI to assist, not replace their strategies. Real-world analogies supported this point: "*When I ask my son to help online, I just want him to read the links I care about, not everything,*" a participant shared. This tension, between helpfulness and overstepping, proved crucial in shaping the next iteration.

Key limitations included:

- No way to interrupt or skip content, which clashed with users' fast-paced, goal-driven behavior
- Lack of granular interaction, such as requesting product variants or navigating modal windows
- Verbose default outputs, which were seen as exhausting and impractical
- Over-reliance on fixed flow, with no user-led pacing, filters, or mode switching

Despite these shortcomings, the prototype generated crucial insights. It demonstrated that natural language interfaces can improve emotional comfort and confidence for BLV users, but only if they respect timing, control, and contextual specificity. These findings validated the need for hybrid systems that combine conversational AI with tactile, interruptible interfaces. As the ASC report's controller paradigm notes, "restoring physical feedback and modality switching" is key to restoring user autonomy (Coppin et al., 2024, Section F.2.5).

4.3 Iteration 2: Rotary-Based Screen Reader Controller

The second design iteration marked a pivotal shift in direction, driven by inductive insights from our initial interview sessions and the first co-design session. Participants testing the conversational assistant frequently expressed the need for faster, more fluid navigation and highlighted how they often skip to the next element as soon as they hear the first word. This behavior pointed toward the importance of a physical, continuous input, something that would allow rapid progression and reversal without cognitive strain. Navigating forward with a screen reader is typically done using the "Tab" or arrow keys, but going backward requires holding "Shift + Tab," which is considered mentally and physically taxing. Similarly, combining heading navigation (“H”) with other keys disrupted flow due to spatial separation on the keyboard. Also, users preferred to continue using their existing screen readers rather than learn entirely new systems. Participants cited familiarity with voice output, shortcut logic, and interface timing as key reasons for preferring established tools. In response, we chose to integrate the prototype with NVDA, one of the most widely used free and open-source screen readers. Its keyboard commands were already well-known to participants, which minimized the need for retraining and allowed us to focus our design efforts on enhancing interaction, rather than replacing the underlying system. By building on an existing screen reader rather than creating a custom speech interface from scratch, we were able to support a steeper learning curve and preserve continuity with users’ everyday workflows.

These insights informed the decision to prototype a physical controller centered around a rotary knob. The action of rotation offered a natural metaphor for moving up and down a list, mirroring volume control or scrolling, while also reinforcing a sense of progress, direction, and momentum. Clockwise and counterclockwise rotation enabled seamless traversal, while single-press actions toggled modes like heading navigation. A participant during the session proposed using a triple-click gesture as a “Back” command. We implemented it on the spot, which led to immediate positive feedback. The familiarity of NVDA was preserved to minimize learning curves, and the result was a unified, tactile interface grounded in user behavior and shaped directly by user-led ideation.

The interaction mapping for this prototype is summarized below:

Table 3. *Interactions of Second Prototype*

User Input	NVDA Command	Action

Rotate Clockwise	Tab	Next Actionable Option
Rotate Counterclockwise	Shift + Tab	Read next focusable item
Single Press	H	Jump to next Heading
Double Press	Enter	Activate item
Triple Press	Alt + Left	Back

A Python listener processed input from the rotary encoder and emulated corresponding keyboard commands.

During co-design testing, participants used the controller to complete both guided tasks and open exploration activities within a web interface. Initial feedback indicated that participants appreciated the compact form factor and the familiarity of rotational gestures. One participant commented, "*It feels like it's somewhere between a trackpad and a trackball.*"

Participants quickly understood the function of clockwise and counterclockwise rotation to navigate forward and backward. The heading jump feature, triggered by a single press, was particularly appreciated for skimming through content efficiently. However, issues emerged around unintended selections during fast rotation, leading to requests for separating heading navigation into a distinct button to avoid input conflict.

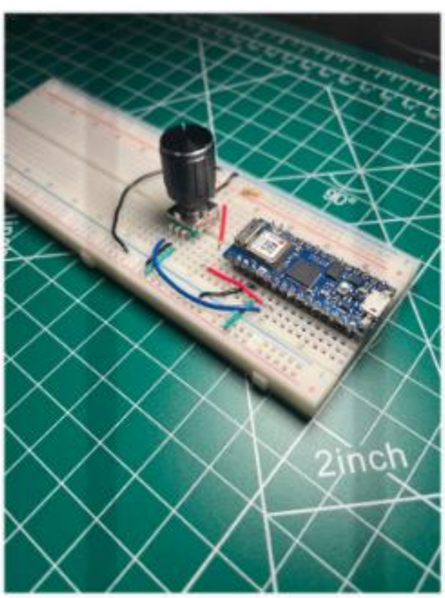


Figure 7. *Left: The screen reader controller prototype built on a breadboard using an Arduino Nano 33 IoT and rotary encoder. Right: A blind participant testing the tactile interaction during a co-design session, using the rotary controller to navigate a web interface via NVDA.*

A recurring theme in the feedback was the absence of multi-directional navigation. Participants emphasized the need for a joystick or similar input to support actions such as navigating by word, character, or line. One participant explained, "*The direction keys would be great. It would mimic a lot of the gestures and a lot of the keystrokes.*" This was reinforced by others who suggested that a joystick metaphor better matched their expectations for fluid and spatial navigation.

While the button-based interaction model was initially seen as intuitive, participants quickly encountered challenges recalling which actions were triggered by which press patterns. The use of double- and triple-clicks for selection and navigation caused confusion; participants often expected a different gesture, or even a separate physical button, for selecting an item. One participant remarked that they were looking for another button to click, revealing that the cognitive load of remembering stacked interactions on a single control outweighed its convenience. These insights suggested that distributing functions across distinct tactile elements, rather than compressing all commands into one knob, might better align with user expectations and reduce mental strain.

Participants also highlighted specific challenges with navigating dropdown menus, checkboxes, and form fields, components often inadequately supported by screen readers. Comments such as "*Dropdowns are ultra annoying*" and "*Form controls like 'I'm not a robot' are never read*" underscored the limitations of the current interaction model. While these issues stem in part from inaccessible web code, something beyond the scope of what a controller alone can address, participants' frustrations revealed a critical need for supplemental support. As designers cannot directly alter HTML structure or browser behavior, this insight motivated a parallel exploration into AI-based solutions (Coppin et al., 2024).

A schematic diagram of a webpage showing various web elements—landmarks in blue, headings in red, focusable items in dark blue, lines in white, and tables in tan. Black arrows indicate knob rotation used to move forward and backward between focusable items. Green arrows represent knob presses used to jump between headings. The diagram demonstrates how the second iteration's controller allows blind users to navigate through structured content using only two interactions: rotation and press. A legend explains each color and interaction symbol.

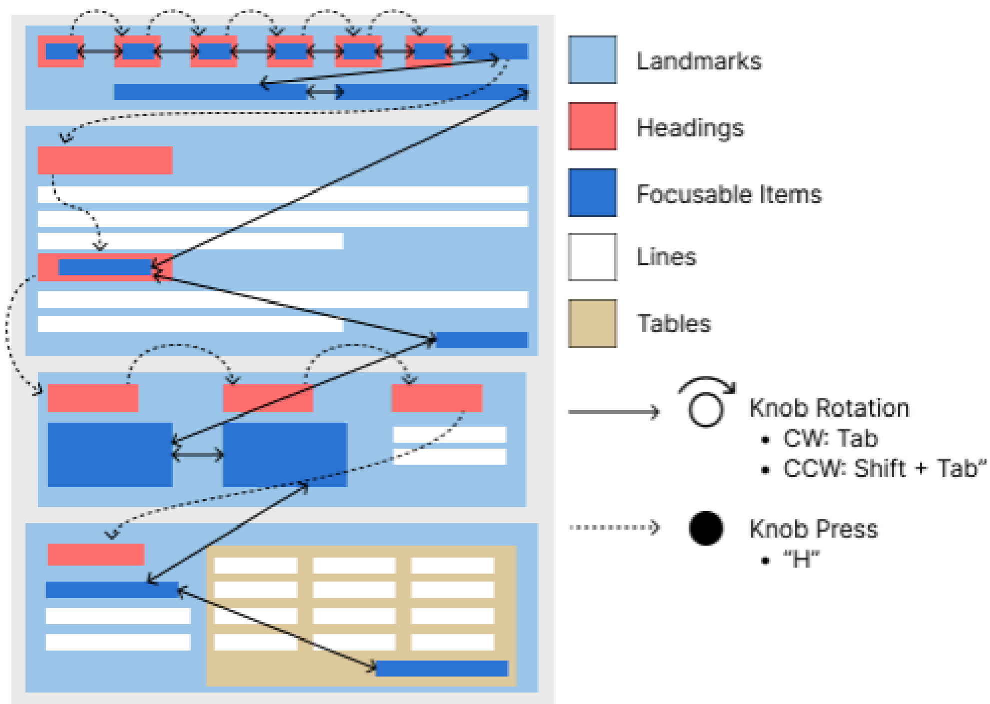


Figure 8. *Infographic illustrating the navigational model of Iteration 2, a rotary-based screen reader controller. This schematic interface represents a typical webpage composed of structural elements such as landmarks, headings, focusable items, lines, and tables. Users rotate the knob to move forward and backward across focusable items, while pressing the knob jumps between headings.*

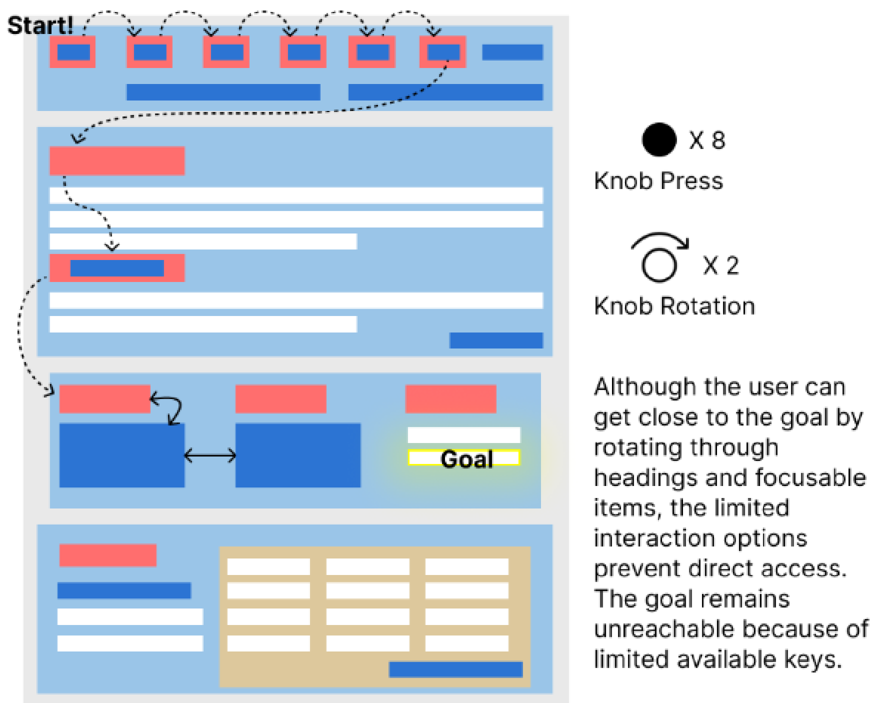


Figure 9. *The user begins at the top of the webpage and navigates through headings and focusable items using 8 knob presses and 2 knob rotations but cannot reach the intended goal due to single-axis limitations.*

One of the most significant insights from this session was the participants' interest in an integrated, on-demand AI assistant. Rather than relying on continuous voice guidance, users wanted the ability to activate assistance only when needed; for example, to clarify an interface element or prompt the system to describe a form. They specifically requested the assistant without disrupting the primary navigation. One user summarized this perspective as follows: "*If I say 'what is this button,' it should tell me, not do something.*"

Finally, there was strong support for a modular design approach. Participants acknowledged that a single controller may not accommodate all needs and expressed openness to expanding the system with additional knobs or joysticks.

4.4 Iteration 3: Modular Controller with Joystick

The third prototype emerged in response to the cognitive, functional, and emotional challenges raised in the first two iterations. While the conversational AI assistant (Iteration 1) offered affective support but lacked responsiveness, and the rotary-only controller (Iteration 2) provided tactile control but insufficient directional and contextual input, this iteration aimed to merge the strengths of both models. A dual-input interface was developed, combining a rotary encoder with a push-button and a multidirectional joystick. An AI assistant was included as a context-aware support feature, triggered through a long press.

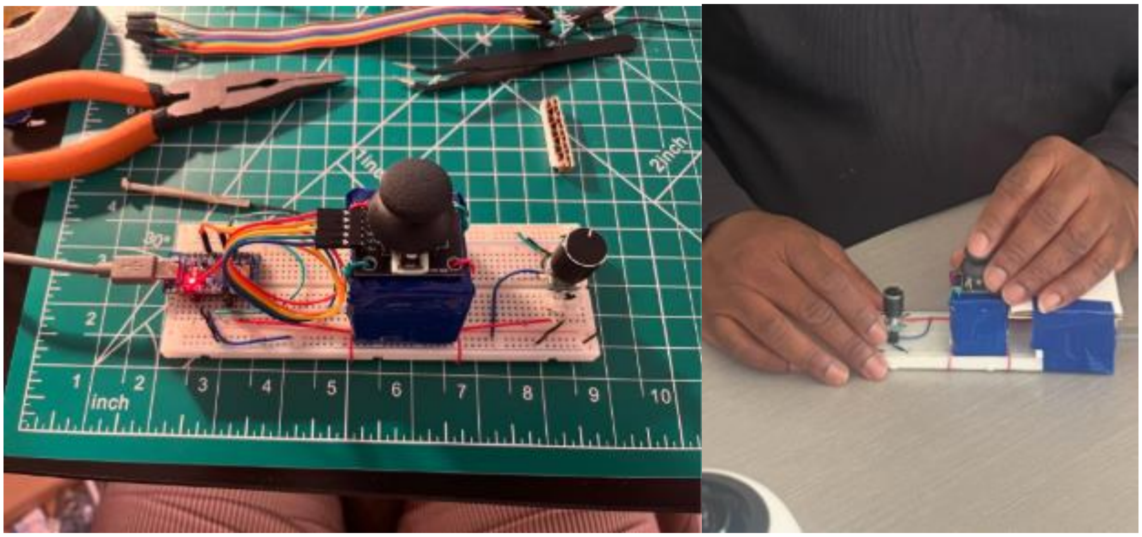


Figure 10. *Prototype of Iteration 3 featuring a rotary knob and joystick on a compact breadboard setup (left), and participant testing the device during a co-design session (right). The design supports tactile navigation, contextual interaction, and AI assistance.*

The goal was to prototype a modular, interruptible navigation model that not only respected user agency but also reintroduced spatial awareness through structured layering. The interaction design translated NVDA shortcuts into tangible controls, aligning with familiar gestures such as swiping, tapping, or rotating through content categories. Notably, the joystick handled continuous arrow key movements (line-by-line, left/right), while the rotary knob was reserved for tab-based traversal (element-by-element). Combined gestures, such as holding the joystick while rotating the knob, triggered heading jumps.

Table 4. *Available Interactions in the Third Prototype*

User Input	NVDA Command	Action
Rotate Knob Clockwise	Tab	Next focusable Item
Rotate Counterclockwise	Shift + Tab	Move backward focusable
Pres and Hold joystick + rotate Knob clockwise	H	Jump to next Heading
Pres and Hold joystick + rotate Knob Counterclockwise	Shift + H	Activate item
Tap knob	Enter	Back
Pres and Hold joystick + Tap Knob	Alt + Left	Back
Tilt joystick (Up/ Down/ Left/ Right)	Arrow (↑/↓/←/→)	Move (Up/ Down/ Left/ Right)
Press and hold knob (>1 sec)	-	Triggers AI assistant

To avoid over-automation, the AI assistant was redesigned to serve as an optional co-pilot rather than a primary narrator. Triggered through a long-press on the knob, the assistant was limited to the current focus area and allowed for follow-up clarifications, "*Tell me only the price and brand,*" one participant suggested. This addressed concerns from earlier iterations that AI guidance often overwhelmed rather than empowered.

"*This one is really cool... it just takes a moment to get used to*," one participant shared when testing the combined controller. They responded positively to the retro-console-like form factor and emphasized the appeal of separating navigation from assistance. Others noted, "*The freedom is great... it mimics a lot of gestures,*" affirming the familiarity of joystick interaction and the alignment with muscle memory.

However, usability challenges remained. Several participants struggled with the cognitive load of remembering layered interactions. The absence of feedback during region transitions led to disorientation, and participants often described feeling lost or unsure of where they were within the interface. In response, many expressed a strong desire for a way to "start fresh" or return to a known point in the interface. This sense of resetting was framed as both a practical and emotional relief, allowing users to regain confidence and proceed more intentionally. As one participant put it, "*Sometimes it's easier to just start over when I feel stuck.*" These reflections informed the inclusion of a dedicated Home button for resetting navigation and a Back button to cancel or undo recent actions. Users requested auditory or haptic signals to anchor spatial context: for instance, a subtle vibration when entering a new section or a chime to indicate reaching a boundary. One participant explained, "*You get to the point where you get a dropdown menu, it would be interesting how it's gonna interpret that*," pointing to unresolved issues with form fields and modals. These limitations revealed the importance of multimodal feedback, not just for navigation, but for reassurance and context awareness (Coppin et al., 2024).

The notion of layering, drawn from smartphone paradigms like the iPhone rotor and explore-by-touch, gained strong traction. The iPhone rotor is a virtual control accessed through a two-finger rotation gesture on the touchscreen, allowing VoiceOver users to quickly switch between navigation modes, such as headings, links, form fields, or custom controls, without leaving their current context. It enables non-visual users to dynamically filter what content the swipe gestures will target next, offering efficient, on-the-fly interaction tailored to their goals. This feature introduced a flexible and layered approach to navigation that influenced many subsequent assistive designs. Participants advocated for a hardware-based selector or mode ring to toggle between content layers (e.g., headings, links, word-level), as well as modular layouts where joystick, knob, and assistant could be repositioned or customized based on task and user needs. These insights informed the design brief for the next step, which will include:

- A tactile mode selector ring for switching navigation types.
- Configurable haptic and auditory cues to support spatial orientation.
- An upgraded AI assistant with voice-based clarification prompts and back-and-forth conversation.
- A modular hardware layout supporting reprogrammable components and ergonomic diversity.

Early feedback from co-design participants suggests this prototype holds strong potential for users who became blind later in life and seek more confidence, particularly because it reduces reliance on memorized shortcuts and enables more intuitive, physically guided interactions. These users often have prior experience with touch-based or visual systems and may find traditional screen readers disorienting due to their command-heavy structure. One participant reflected, "*Normally when I go through my phone or iPad, I'm swiping. So this actually saves a little bit of time*," affirming the system's ability to blend familiarity with new affordances.

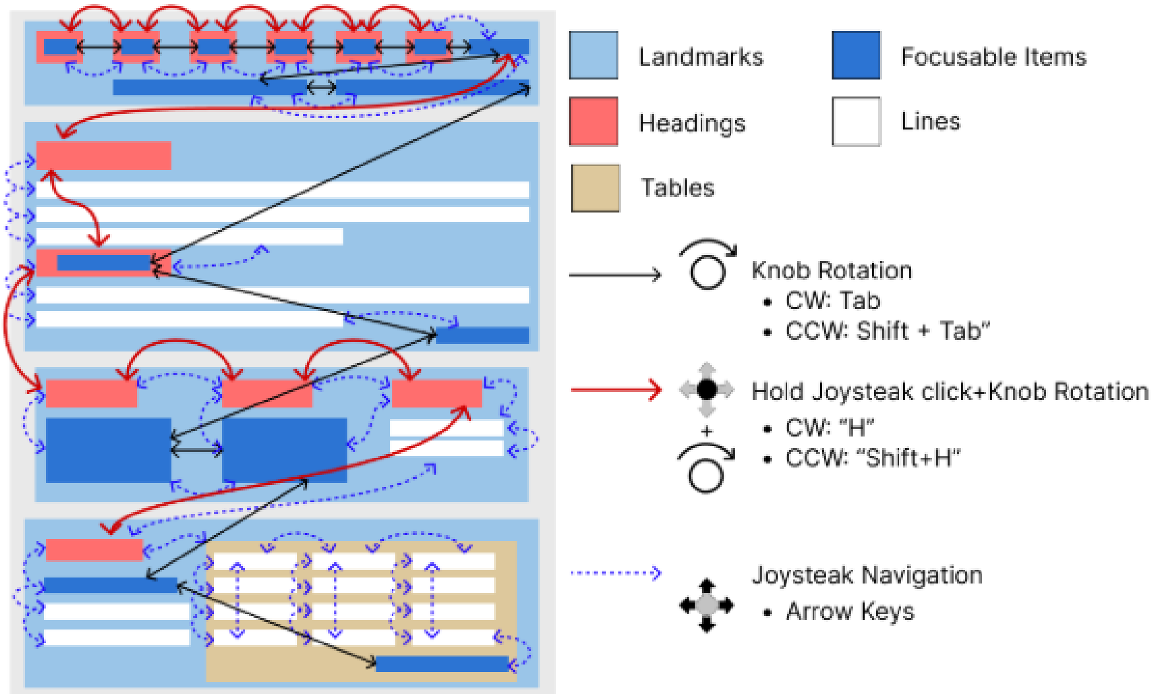


Figure 11. *A dual-input screen reader controller combining rotary knob and joystick inputs. This schematic interface represents a typical webpage composed of elements such as landmarks, headings, focusable items, lines, and tables. Users rotate the knob to navigate forward or backward across focusable items, while holding the joystick and rotating the knob enables heading-based traversal. The joystick allows granular, multidirectional movement across content, enhancing control over spatial navigation within complex web layouts.*

The third prototype validated the need for hybrid, multi-modal interaction, neither fully linear nor fully conversational. It reinforced that assistive tools for blind users must balance control and support, while offering a sense of orientation, predictability, and freedom.

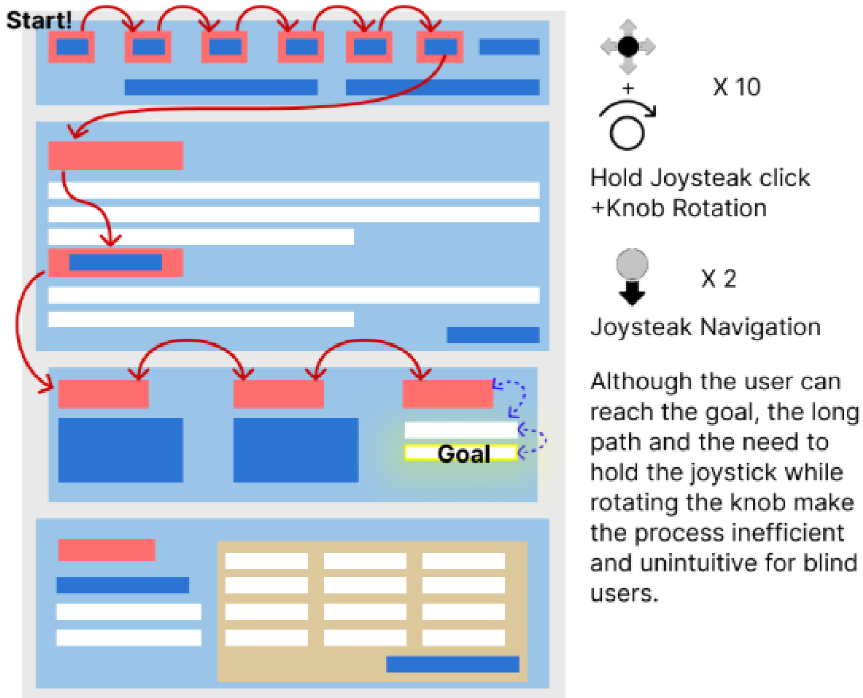


Figure 12. *The user begins at the top of the webpage and reaches the goal by combining 10 joystick+knob rotations and 2 joystick tilted to down. While the task is technically achievable, the long path and need to coordinate two inputs make the interaction inefficient and unintuitive for blind users.*

4.5 Final Design Suggestion

The final iteration of this project culminated in a modular, affordance-driven controller designed to enhance digital accessibility for blind and low-vision (BLV) users. Developed through iterative prototyping, interviews, and co-design sessions, the interface augments traditional screen reader workflows by introducing tactile input methods that support spatial awareness, emotional reassurance, and cognitive clarity.

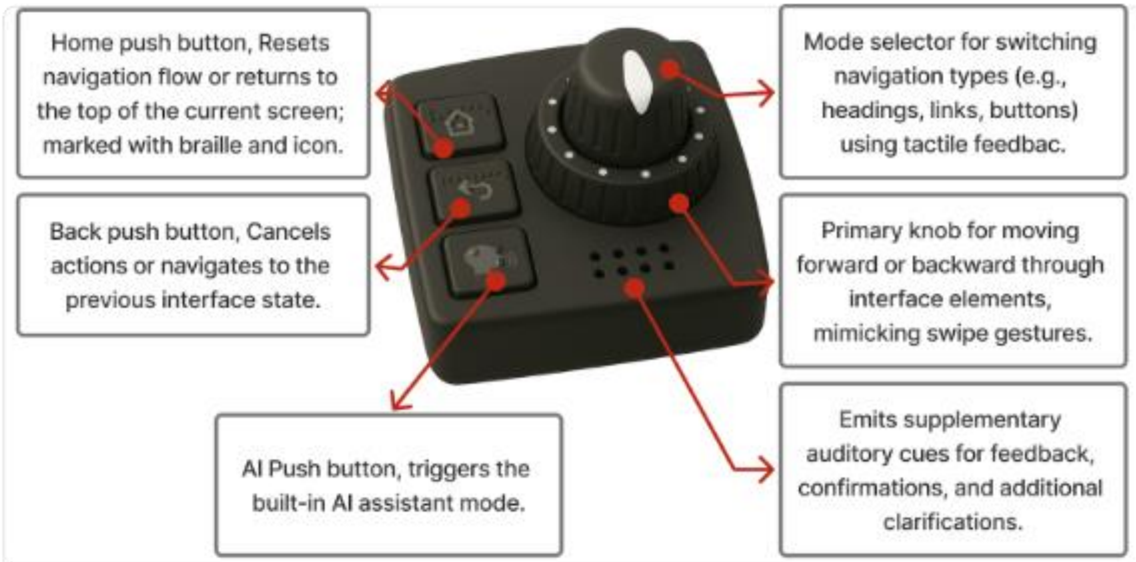


Figure 13. *Annotated breakdown of the controller's physical interface, highlighting key interactive components including the rotor, rotary knob, AI trigger, home and back buttons, and audio feedback output. Each element is designed to support non-visual digital navigation through tactile, auditory, and context-aware feedback.*

At the core of the device is a rotary knob that facilitates linear traversal through interface elements. Rotating the knob clockwise or counterclockwise allows users to move forward or backward through content, offering a more fluid and continuous experience compared to the abrupt, disjointed nature of keyboard commands. Several participants noted that this interaction made content navigation feel more "mentally graspable" and allowed for a rhythm that supported focused browsing.

The second core feature is an analog rotor switch designed to emulate the function of the iPhone rotor. This selector allows users to shift between navigation modes such as headings, links, and form elements without relying on shortcut memorization. Each notch of the selector physically confirms a mode change and reinforces the sense of user control. One participant described it as "*changing gears*" in the navigation logic, appreciating the ability to align their strategy with the task at hand, whether skimming content or targeting interactable components.

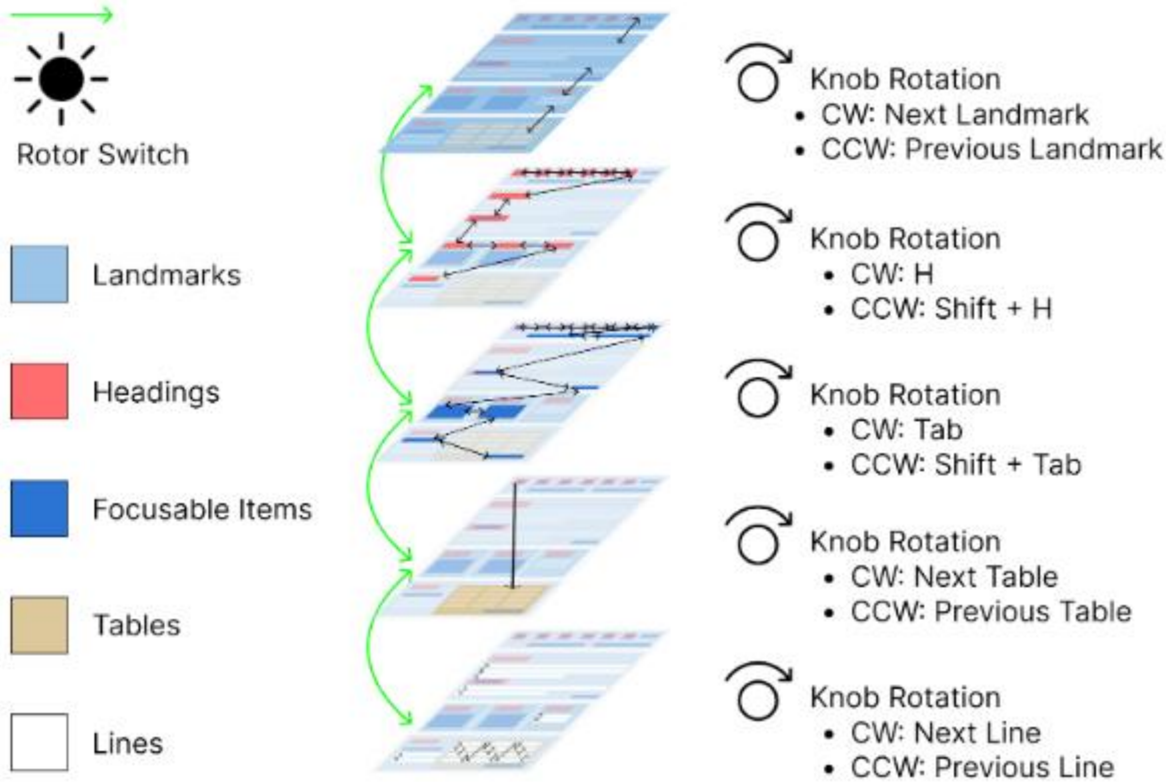


Figure 14. *The screen reader controller introduces a rotor switch that allows users to select between semantic layers, landmarks, headings, focusable items, tables, and lines, each providing a dedicated navigation mode. Once a mode is selected, the user can rotate the main knob clockwise or counterclockwise to move forward or backward through elements within that layer. This layered structure addresses previous limitations by enabling multi-axis control and precise, context-aware navigation through a website's structure.*

This version also introduced haptic and auditory feedback to reinforce actions and transitions. A short vibration indicates successful region entry or completion, while a sharper tone signals navigation errors or boundaries. These cues provide grounding during interaction, helping users maintain orientation without requiring excessive verbal feedback.

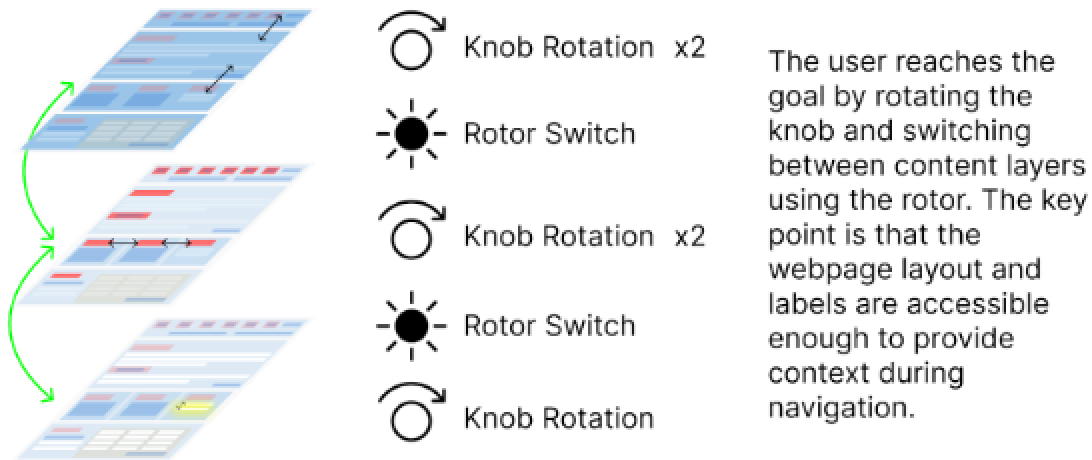


Figure 15. *To reach the goal, the user rotates the knob and switches between content layers using the rotor. This layered interaction path assumes the webpage layout and element labels are sufficiently accessible to provide navigational context.*

The AI assistant button that changes the mode of the navigation to a voice assistant, which is aware of the cursor location, is designed to offer optional, context-aware support. Unlike earlier prototypes that risked dominating the interaction, this assistant prioritizes region-specific summaries and back-and-forth clarification. Its responses are grounded in the user's current DOM (Document Object Model) focus and surrounding HTML context (Coppin et al., 2024). Users can ask follow-up questions to retrieve only the information they need, such as "Tell me what color this product is." One participant stated, "I feel like I'm missing something. I just want it to confirm what I'm trying to find," reinforcing the importance of concise, user-controlled responses. This feature also replaces the previously manual process of sending screen captures to third-party tools such as Be My Eyes, providing immediate in-device support.

The controller also includes a Home button and a Back or Cancel button. The Home button, marked with a tactile braille label, allows users to reset the interaction or return to the top of a page. The Back button is designed to cancel current actions, close windows, or return to the previous screen. Both buttons were shaped in response to participant requests for fast, dependable escape mechanisms during complex interactions. These buttons provide clearly defined affordances and tactile feedback, helping users avoid confusion or unintended actions.

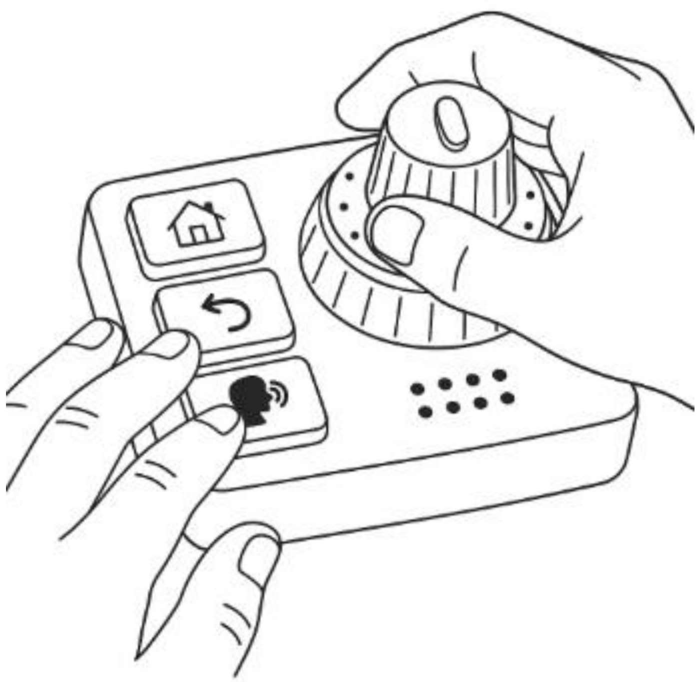


Figure 16. *Schematic illustration of the final controller design, showing an ergonomic, handheld form factor sized for two-handed use. The layout supports comfortable thumb access and minimizes finger movement, reducing fatigue during extended non-visual navigation.*

5. DISCUSSION

Screen reader interaction is shaped by several interdependent elements that influence how blind and low-vision (BLV) users perceive, navigate, and interact with digital content. Drawing from accessibility and human-computer interaction (HCI) literature, this study identifies six core experience dimensions. These dimensions are used to assess the final design in terms of physical, perceptual, and cognitive affordances.

5.1 Accessible Content & Semantic Structure

Semantic HTML elements, such as proper headings, landmarks, and alt text, enable screen readers to convey structure meaningfully through auditory output (Giudice et al., 2020). Without structured markup, BLV users must process content linearly, resulting in disorientation and effortful navigation (Information Wayfinding of Screen Reader Users, 2022). While the controller cannot directly modify web content, it builds upon these structures through tactile tools that help users traverse semantically meaningful elements like links and headings with precision and control.

5.2 Layout Complexity & Navigation Map

Disorganized or visually complex interfaces, especially those with hidden content or dynamic elements, exacerbate cognitive load when presented through screen readers (User Experience Study of Screen Readers, 2023). To address this, the controller supports mode-switching via an analog rotor, allowing users to filter content layers based on type (e.g., headings, form fields), thereby reducing the need for exhaustive linear traversal. This design echoes the benefits of "skip links" and other structural shortcuts, offering a physical mechanism to map and mentally anchor the interface.

5.3 Input Modalities & Interaction Methods

The system replaces the complexity of keyboard combinations (e.g., Shift+Tab or Insert+F7) with accessible tactile gestures. The rotary knob supports continuous, bidirectional movement, and the rotor switch enables analog mode selection, mimicking touch gestures like flicks or iPhone rotor twists. These modalities were preferred by participants for their ease of use, reduced hand strain, and alignment with muscle memory developed through previous mobile or gaming interfaces.

5.4 Output Modalities & Feedback Mechanisms

Feedback in this prototype is intentionally multimodal. Supplementary audio cues signal key system events (e.g., page loads, error states) using minimalist earcons that avoid auditory overload. Complementary vibration patterns reinforce spatial cues such as reaching a page end or confirming a selection. This aligns with Akkaya et al.'s (2023) recommendation for real-time, bidirectional haptic feedback systems that dynamically respond to user intent. These outputs enhance perceptual awareness while preserving the user's attention and reducing uncertainty.

5.5 Customization & Personalization Settings

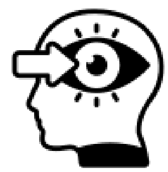
While full reconfiguration options were outside the scope of this prototype, users can adjust their experience through interaction pacing. The speed of knob rotation, the frequency of AI assistance, and the choice to use physical buttons (e.g., Home or Back) reflect individual preference and task demands. This flexibility mirrors McCarthy et al.'s (2013) insights on verbosity, gesture assignments, and speech rate control as tools for managing cognitive load and comfort.

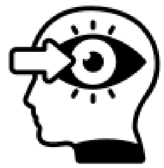
5.6 User Skills, Experience, and Strategies

Over time, users developed procedural memory for different interaction combinations, such as rotating the knob to skim, pressing the rotor to change layers, or calling on AI for clarification. These habits reflect the strategic adaptation emphasized by Borodin et al. (2010), where expert users blend tools and mental models to navigate complex tasks. The physical layout of the controller supports this skill development by offering discrete, easy-to-locate components that build confidence and efficiency with repeated use

5.7 Summary

Table 5. *Affordance-Based Evaluation of the Final Design Across Six Core UX Dimensions*

	Physical Affordances 	**Perceptual Affordances** 	**Cognitive Affordances** 
Accessible Content & Semantic Structure	The rotary knob and joystick enable direct navigation via screen reader-compatible shortcuts (e.g., Tab, H) without complex keyboard layouts, offering physical precision through continuous, tactile interaction.	Spoken feedback and mode announcements translate the semantic structure into auditory cues, enabling non-visual recognition of page elements like headings and lists	Structured interaction with heading lists or labeled controls reinforces memory and supports skimming by building mental models of the interface.
Layout Complexity & Navigation Map	The rotor switch provides mode changes for navigating layers like headings or links, minimizing repeated inputs and supporting fast orientation with analog motion.	Supplementary audio cues and vibration signals confirm mode changes and interface boundaries, reinforcing layout awareness without visual context.	Layered navigation supports mental mapping by separating structure from content, allowing users to anticipate and revisit key regions efficiently.
Input Modalities & Interaction Methods	The hardware includes press, rotate, and toggle inputs, mapping screen reader actions to gestures familiar from touch and gaming interfaces, offering motor-	Feedback tones and speech confirm user actions in real-time, helping users perceive the impact of each gesture or button press with confidence.	The analog control model mimics familiar tools (e.g., rotor, arrow keys), making it easier to learn and remember through embodied repetition.

	friendly access without keyboards or touchscreens.		
	Physical Affordances 	**Perceptual Affordances** 	**Cognitive Affordances** 
Output Modalities & Feedback Mechanisms	Built-in speaker and haptic motor provide immediate, low-effort feedback via earcons and vibrations, making responses physically perceivable through multiple sensory channels.	Earcons (e.g., page load, error, confirmation) and distinct vibration patterns ensure responses are perceptually clear without overwhelming auditory load.	Consistent feedback patterns (sound and haptics) help users track progress and detect status changes, supporting error recovery and spatial orientation.
Customization & Personalization Settings	Users can adjust interaction speed through rotation pace or rely on alternate controls like dedicated buttons, adapting the physical experience to their personal preferences and motor strengths.	Output varies in intensity and detail depending on input speed or context, allowing users to personalize their experience based on sensory preference.	The system offers multiple paths to the same goal (e.g., knob, AI, headings), letting users adopt and refine strategies that match their experience and needs.
User Skills, Experience, & Strategies	The layout supports quick mastery; over time users become fluent in tactile sequences, knob resistance, and spacing between controls, reducing reliance on external aids or shortcuts.	Users develop perceptual fluency over time, quickly recognizing output tones, vibrations, and speech variations linked to system states or navigation modes.	Skilled users can combine interaction modesâ€”physical, auditory, and AIâ€”to plan, execute, and adjust their tasks with adaptive,

Each of the final prototype’s features was co-designed with BLV users and grounded in existing screen reader behaviors while extending interaction into a more tangible and expressive modality.

The rotary knob, for instance, provides not only precise movement through content but also an embodied metaphor for control and scanning. Its physical resistance and continuous motion allow users to navigate rhythmically, mirroring how sighted users visually skim.

The rotor switch offers a parallel to the iPhone rotor, enabling mode-switching in a physically intuitive form. Participants described the satisfaction of “feeling” through the interface, a strategy that enhanced both spatial orientation and confidence. The inclusion of Home and Back buttons emerged from co-design feedback around disorientation. These buttons serve as cognitive anchors, giving users easy paths to reset or retreat, supporting the ASC report’s (Coppin et al., 2024) emphasis on “cooperative control.”

The AI assistant, activated by a dedicated button, adds an optional layer of conversational support. Participants compared it to asking a sighted person to "just read the part I care about." This preference reflects broader research by Hegde (2023), which found that blind users favor AI systems that assist rather than direct. The assistant here never takes control, but instead offers summaries or clarifications on request, making it a trusted partner rather than an interruptive narrator.

Minimalist audio cues and vibration feedback were designed based on the perceptual limits identified by Fink et al. (2023), who noted that complex spatial audio menus can be error-prone and fatiguing. By using familiar signals (e.g., checkmark chimes, short buzzes), the prototype confirms without adding processing burden. These layered cues support both perceptual clarity and spatial awareness, particularly in dynamic or form-heavy environments.

Finally, this research contributes a meaningful step toward enabling non-visual skimming, a behavior frequently cited as lacking in current screen reader paradigms (Bigham et al., 2007, 2017; Bragg et al., 2018). The rotary knob allows for rapid traversal of structured content, while the rotor switch and optional AI assistant support different levels of depth and granularity. This hybrid approach—tactile, auditory, and AI-assisted—enables users to move fluidly between broad overviews and targeted actions, a dynamic previously only accessible to sighted users through visual scanning. Ahmad et al. (2012) introduced early work on automated skimming, but this prototype extends that.

Taken together, the prototype offers more than a hardware accessory, it proposes a shift in how non-visual interaction can be designed. By attending to the full range of physical, perceptual, and cognitive affordances across six core UX dimensions, the controller provides BLV users with a system that is intuitive, adaptable, and user-directed. It not only addresses known gaps in accessibility tools but opens new possibilities for how non-visual skimming, layered control, and multimodal feedback can converge in a single, ergonomic interface. This moves the conversation beyond compliance and toward a vision of fluent, confident, and expressive screen reader navigation.

6. CONCLUSION

This study demonstrated that current screen reader systems impose significant cognitive, spatial, and emotional burdens on blind and low-vision (BLV) users, stemming largely from linear interaction models and the absence of contextual and paralinguistic feedback. Through rigorous human-centered research, including interviews, co-design, and usability testing, a novel, modular controller was developed that combines tactile inputs (rotary knob and joystick), mode-switching, and AI-driven assistance to support layered, non-linear navigation. Findings highlight the value of tangible, customizable tools in enhancing user agency, efficiency, and orientation. However, challenges remain in ensuring intuitive AI integration without over-automation and in addressing the trade-offs between control complexity and cognitive load. Future iterations should refine multimodal feedback mechanisms and explore personalization in AI behavior based on user intent and task context.

7. REFERENCES

1. Gaver, W. W. (1991, March). Technology affordances. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 79–84). https://doi.org/10.1145/108844.108856
2. Coppin, P., Hung, P., Ingino, R., Quevedo, A. U., Sukhai, M., & Syed, A. R. (2024, March). A study of accessible and inclusive virtual and blended information and communication technologies (ICTs) for the federal public service and federally regulated industries in post-COVID-19 Canada. Accessibility Standards Canada. https://sites.google.com/view/asc-virtual-icts-accessibility/
3. Bigham, J. P., Lin, I., & Savage, S. (2017, October). The effects of "not knowing what you don't know" on web accessibility for blind web users. In Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility (pp. 101–109). https://doi.org/10.1145/3132525.3132543
4. Larkin, J. H., & Simon, H. A. (1987). Why a diagram is (sometimes) worth ten thousand words. Cognitive Science, 11(1), 65–100. https://doi.org/10.1111/j.1551-6709.1987.tb00863.x
5. Lazar, J., Allen, A., Kleinman, J., & Malarkey, C. (2007). What frustrates screen reader users on the web: A study of 100 blind users. International Journal of Human-Computer Interaction, 22(3), 247–269. https://doi.org/10.1080/10447310709336964
6. Murphy, H. C., Hofacker, C. F., & Mizerski, R. (2008). Priming and the dimensions of website effectiveness. Journal of Advertising, 37(1), 49–61. https://doi.org/10.2753/JOA0091-3367370104

7. Williams, M., Ferguson, R., & Kirkham, R. (2019). Digital accessibility for disabled people: A cultural critique. Disability & Society, 34(10), 1602–1622. https://doi.org/10.1080/09687599.2019.1619234

8. Lagman, E. (2016). The impact of screen reader limitations on blind users' online emotional engagement. Journal of Assistive Technologies, 10(3), 178–190.

9. Ahmad, F., Borodin, Y., Soviak, A., Islam, M., Ramakrishnan, I. V., & Hedgpeth, T. (2012, October). Accessible skimming: Faster screen reading of web pages. In Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (pp. 367–378). https://doi.org/10.1145/2380116.2380165

10. Dissanayake, D. (2015). Visualizing structure for screen readers: Web page preview design to support mental mapping. Unpublished master's thesis, University of Victoria.

11. Baez, M., Cutrupi, C. M., Matera, M., Possaghi, I., Pucci, E., Spadone, G., … & Pasquale, A. (2022, April). Exploring challenges for conversational web browsing with blind and visually impaired users. In CHI Conference on Human Factors in Computing Systems Extended Abstracts (pp. 1–7). https://doi.org/10.1145/3491101.3519720

12. Zhang, L., Sun, S., & Findlater, L. (2023, October). Understanding digital content creation needs of blind and low vision people. In Proceedings of the 25th International ACM SIGACCESS Conference on Computers and Accessibility (pp. 1–15). https://doi.org/10.1145/3597638.3608451

13. Hegde, M. (2023). User experience study of screen readers for visually challenged users (Master's thesis, Purdue University). https://hammer.purdue.edu/articles/thesis/User_Experience_Study_of_Screen_Readers_for_Visually_Challenged_Users/24150301

14. Bragg, D., Bennett, C., Reinecke, K., & Ladner, R. (2018, April). A large inclusive study of human listening rates. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (pp. 1–12). https://doi.org/10.1145/3173574.3173824

15. Choi, D., Kwak, D., Cho, M., & Lee, S. (2020, April). “Nobody speaks that fast!”: An empirical study of speech rate in conversational agents for people with vision impairments. In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (pp. 1–13). https://doi.org/10.1145/3313831.3376634

16. Graham, G. (2015). A process and tool for evaluation of accessible user experience with websites and apps for screen reader users who are blind. Inclusive Design Research Centre.

17. Jordan, J. B., Van Hyning, V., Jones, M. A., Montgomery, R. B., Bottner, E., & Tansil, E. (2024, October). Information wayfinding of screen reader users: Five personas to expand conceptualizations of user experiences. In Proceedings of the 26th International ACM SIGACCESS Conference on Computers and Accessibility (pp. 1–7). https://doi.org/10.1145/3611033.3614262

18. Borodin, Y., Bigham, J. P., Dausch, G., & Ramakrishnan, I. V. (2010, April). More than meets the eye: A survey of screen-reader browsing strategies. In Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A) (pp. 1–10). https://doi.org/10.1145/1805986.1805991

19. Vtyurina, A., Fourney, A., Morris, M. R., Findlater, L., & White, R. W. (2019). Bridging screen readers and voice assistants for enhanced eyes-free web search. In The World Wide Web Conference (pp. 3590–3594). https://doi.org/10.1145/3308558.3314136

20. Fink, P. D., Abou Allaban, A., Atekha, O. E., Perry, R. J., Sumner, E. S., Corey, R. R., … & Giudice, N. A. (2023, March). Expanded situational awareness without vision: A novel haptic interface for use in fully autonomous vehicles. In Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction (pp. 54–62). https://doi.org/10.1145/3568294.3580084

21. Giudice, N. A., Guenther, B. A., Jensen, N. A., & Haase, K. N. (2020). Cognitive mapping without vision: Comparing wayfinding performance after learning from digital touchscreen-based

multimodal maps vs. embossed tactile overlays. Frontiers in Human Neuroscience, 14, 87. https://doi.org/10.3389/fnhum.2020.00087

22. McCarthy, T., Pal, J., & Cutrell, E. (2013). The "voice" has it: Screen reader adoption and switching behavior among vision impaired persons in India. Assistive Technology, 25(4), 222–229. https://doi.org/10.1080/10400435.2013.768719

23. Ghodke, U., Yusim, L., & Somanath, S. (2019). The cross-sensory globe: Co-designing a 3D audio-tactile globe prototype for blind and low-vision users to learn geography. In Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility (pp. 361–363). https://doi.org/10.1145/3308561.3353785

8. APPENDICES

8.1 appendix A- Arduino Code Iteration 2

```
    #include <Rotary.h>
#include <Button2.h>
#include <Keyboard.h>

// === Pin Setup ===
#define ROTARY_PIN1 2
#define ROTARY_PIN2 3
#define ROTARY_BUTTON_PIN 4
#define JOY_BUTTON_PIN 5
#define JOY_VRX A1
#define JOY_VRY A0

// === Joystick Thresholds ===
#define JOY_CENTER 512
#define DEADZONE 100
#define REPEAT_DELAY 100  // ms between repeated arrow key sends
      hhhhhhhhhHHHHHHhhhhh
// === Objects ===
Rotary r;
Button2 rotaryButton;
Button2 joyButton;
```

```
unsigned long lastRepeatTime = 0;
bool joystickHeld = false;

void setup() {
  Serial.begin(9600);
  Keyboard.begin();

  // Rotary setup
  r.begin(ROTARY_PIN1, ROTARY_PIN2, 4, -9999, 9999, 0, 10);
  r.setLeftRotationHandler(onRotateLeft);
  r.setRightRotationHandler(onRotateRight);

  // Buttons
  rotaryButton.begin(ROTARY_BUTTON_PIN);
  rotaryButton.setTapHandler(onRotaryPress);

  joyButton.begin(JOY_BUTTON_PIN);
  joyButton.setPressedHandler(onJoystickHeld);
  joyButton.setReleasedHandler(onJoystickReleased);
}

void loop() {
  r.loop();
  rotaryButton.loop();
  joyButton.loop();
  handleJoystickMovement();
}

// === Rotary Rotation Logic ===
void onRotateRight(Rotary& r) {
  if (joystickHeld) {
    Serial.println("Held + CW → H");
    Keyboard.press('h');
    delay(10);
    Keyboard.release('h');
  } else {
    Serial.println("→ Tab");
    Keyboard.press(KEY_TAB);
    delay(10);
    Keyboard.release(KEY_TAB);
  }
}

void onRotateLeft(Rotary& r) {
  if (joystickHeld) {
    Serial.println("Held + CCW → Shift + H");
    Keyboard.press(KEY_LEFT_SHIFT);
    Keyboard.press('h');
    delay(10);
    Keyboard.release('h');
    Keyboard.release(KEY_LEFT_SHIFT);
  } else {
    Serial.println("← Shift + Tab");
    Keyboard.press(KEY_LEFT_SHIFT);
    Keyboard.press(KEY_TAB);
    delay(10);
    Keyboard.release(KEY_TAB);
    Keyboard.release(KEY_LEFT_SHIFT);
```

```
  }
}

// === Rotary Button Press ===
void onRotaryPress(Button2& btn) {
  if (joystickHeld) {
    Serial.println("Rotary Press + Joystick Held → Alt + Left (Back)");
    Keyboard.press(KEY_LEFT_ALT);
    Keyboard.press(KEY_LEFT_ARROW);
    delay(10);
    Keyboard.release(KEY_LEFT_ARROW);
    Keyboard.release(KEY_LEFT_ALT);
  } else {
    Serial.println("Rotary Press → Enter");
    Keyboard.press(KEY_RETURN);
    delay(10);
    Keyboard.release(KEY_RETURN);
  }
}

// === Joystick Held State Handlers ===
void onJoystickHeld(Button2& btn) {
  Serial.println("Joystick Pressed → Modifier Mode ON");
  joystickHeld = true;
}

void onJoystickReleased(Button2& btn) {
  Serial.println("Joystick Released → Modifier Mode OFF");
  joystickHeld = false;
}

// === Joystick Movement to Arrow Keys ===
void handleJoystickMovement() {
  int x = analogRead(JOY_VRX);
  int y = analogRead(JOY_VRY);
  unsigned long now = millis();

  if (now - lastRepeatTime < REPEAT_DELAY) return;

  if (x < JOY_CENTER - DEADZONE) {
    Serial.println("Joystick → LEFT");
    Keyboard.press(KEY_LEFT_ARROW);
    delay(10);
    Keyboard.release(KEY_LEFT_ARROW);
    lastRepeatTime = now;
  } else if (x > JOY_CENTER + DEADZONE) {
    Serial.println("Joystick → RIGHT");
    Keyboard.press(KEY_RIGHT_ARROW);
    delay(10);
    Keyboard.release(KEY_RIGHT_ARROW);
    lastRepeatTime = now;
  }

  if (y < JOY_CENTER - DEADZONE) {
    Serial.println("Joystick → UP");
    Keyboard.press(KEY_UP_ARROW);
    delay(10);
    Keyboard.release(KEY_UP_ARROW);
```

```
  lastRepeatTime = now;
 } else if (y > JOY_CENTER + DEADZONE) {
  Serial.println("Joystick → DOWN");
  Keyboard.press(KEY_DOWN_ARROW);
  delay(10);
  Keyboard.release(KEY_DOWN_ARROW);
  lastRepeatTime = now;
 }
}
```

8.2 appendix A- Python Code Iteration 2

```
import serial import keyboard import time

PORT = "COM7" # Adjust your port here BAUD = 9600

ser = serial.Serial(PORT, BAUD, timeout=1)

print("Listening for rotary encoder commands...")

click_buffer = [] last_click_time = 0 click_timeout = 0.4 # Time to wait before evaluating click count

while True: try: line = ser.readline().decode('utf-8').strip() now = time.time()

  # Handle rotation immediately
  if line == "tab":
     keyboard.press_and_release('tab')
     print("tab")

  elif line == "shift_tab":
     keyboard.press('shift')
     keyboard.press_and_release('tab')
     keyboard.release('shift')
     print("shift_tab")

  # Handle H click (single/double/triple)
  elif line == "H":
     click_buffer.append(now)
     last_click_time = now

  # Check for click pattern timeout
  if click_buffer and (now - last_click_time > click_timeout):
     count = len(click_buffer)

     if count == 1:
        keyboard.press_and_release('h')
        print("Single Click → H")

     elif count == 2:
        keyboard.press_and_release('enter')
        print("Double Click → ENTER")

     elif count >= 3:
        keyboard.press('alt')
        keyboard.press_and_release('left')
```

```
        keyboard.release('alt')
        print("Triple Click → ALT + LEFT")

    # Clear buffer for next round
    click_buffer.clear()

except Exception as e:
    print("Error:", e)
```