

ABSTRACT

Title of thesis: TANGIBLE WEB LAYOUT DESIGN FOR
BLIND AND VISUALLY IMPAIRED PEOPLE

Ashrith Shrinivas Shetty
Master of Science, 2020

Thesis directed by: Dr. Ge Gao
College of Information Studies

Although past research has enabled blind and visually impaired (BVI) developers to access information better and code more efficiently, they still lack accessible ways to create visual layouts. This research highlights the potential of using a tangible user interface (TUI) to enable BVI people to design web layouts, and presents *Sparsha*, a novel TUI for layout design.

I conducted a semi-structured interview and a co-design session with a blind participant. Based on the elicited insights and designs, I implemented *Sparsha* using 3D-printed tactile elements, an Arduino-powered sensing circuit and a Web server that renders the final HTML layout.

Users place tactile beads on a base to represent HTML elements on the screen. The Arduino senses the type and location of these beads and sends it to a web server which renders the correct HTML element in the correct location on the client browser, and provides audio feedback.

TANGIBLE WEB LAYOUT DESIGN FOR BLIND AND VISUALLY IMPAIRED PEOPLE

by

Ashrith Shrinivas Shetty

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2020

Advisory Committee:

Dr. Ge Gao, Chair/Advisor

Dr. Huaishu Peng, Advisor

Dr. Niklas Elmqvist

© Copyright by
Ashrith Shrinivas Shetty
2020

Acknowledgments

I owe my gratitude to all the people who have made this thesis possible and made my time here, at the University of Maryland, an enriching and fulfilling experience.

I'd like to thank my advisor, Dr.Huaishu Peng, for giving me an invaluable opportunity to work under his guidance. He was always generous with his advice and guidance and I am thankful for the opportunity to work in the Technical HCI lab. It has been a pleasure to work with him and I have learnt a lot.

I would like to extend special thanks to Dr.Ge Gao for stepping in to be my official advisor and the chair of this committee, as well as Dr. Niklas Elmqvist for agreeing to be on my committee.

My colleagues at the Technical HCI lab have been a great source of help and advice for this project and deserve a special mention for all that I could learn from them.

I want to heartily thank all my friends, both in the US and back home in India, who have always been there for me.

I would also like to thank my parents and my brother Ankith for their unwavering support.

Table of Contents

Acknowledgements	ii
Table of Contents	iii
List of Figures	v
1 Introduction	1
1.1 Motivation: From Consumers to Creators	1
1.2 The Tangible Approach	3
1.3 Research Questions	3
1.4 Contributions	4
1.5 Thesis Overview	4
2 Related Work	6
2.1 Tangible User Interfaces	6
2.2 TUI Applications and Efficacy	7
2.2.1 Learning	7
2.2.2 Programming	8
2.2.3 Accessibility for BVI Users	9
2.2.4 Tactile Graphics	10
2.3 Visual Layouts for BVI developers	10
2.4 Summary	11
3 Formative Study	13
3.1 Introduction	13
3.2 Interview procedure	13
3.3 Background	13
3.4 Interview Themes	14
3.4.1 Use of assistive technologies	14
3.4.2 Web development experience	15
3.4.3 Opinions on learning	16
3.5 Key Findings and Design Implications	16
4 Design Considerations and Participatory Design	18
4.1 Design Criteria and Solutions	18
4.2 Design of Tactile Elements	20

4.3	What is Participatory Design?	20
4.4	Design Session Goal	21
4.5	Artifacts	21
4.6	Design Session Process	22
4.7	Results and Observations	24
5	Sparsha: A Tangible Layout Design Tool	28
5.1	System Overview	28
5.2	Electronics	29
5.3	Chess Piece Identification Technology	30
5.3.1	Key Switch Matrix	31
5.3.2	Matrix of Ohmmeters: Bead Type Recognition	33
5.4	Tactile elements	36
5.4.1	Base board	36
5.4.2	Beads and edge constraint design	37
5.5	Web Application	38
5.6	Bringing it all together	39
6	Summative Evaluation Plan	41
6.1	Participants	41
6.2	Method	42
6.2.1	Stage 1: Screening and Introduction	42
6.2.2	Stage 2: Tasks	43
6.2.2.1	Data Collection and Evaluation Metrics	43
6.2.3	Stage 3: Interview and discussion	44
7	Discussion and Conclusion	45
7.1	Limitations	45
7.2	Future Work	46
7.3	Conclusion	47
	Appendix A Interview Script	48
	Appendix B Screening survey	50
	Bibliography	51

List of Figures

4.1	The various artifacts used for the co-design session.	21
4.2	User interacting with one of the base artifacts during the co-design session.	23
4.3	The pillared base design	24
4.4	The user preferred bead and edge constraint design	25
4.5	The selected bead designs for tactile texture	27
5.1	Sparsha: System Diagram	28
5.2	3x3 version of the key switch matrix circuit with diodes.	32
5.3	Switching Diodes being soldered to the 12x6 PCB	32
5.4	Calculating resistance using a voltage divider circuit	33
5.5	Circuit diagram of a 3x3 version with diodes and reference resistors.	34
5.6	PCB after soldering the diodes and resistors	35
5.7	The base board design with magnets and copper tape forming the connections	36
5.8	2 pin connectors were used to make connections from the PCB to the base board	37
5.9	The 12 column CSS grid container used on the client (if the outlines were visible)	39

Chapter 1: Introduction

1.1 Motivation: From Consumers to Creators

Advances in accessible and assistive technologies have simplified the lives of blind and visually impaired (BVI) people by enabling them to not only consume digital content but also to turn into digital creators and social media influencers [2, 3, 9]. *Supporting Blind Photography* [36] makes the case for blind people’s desires to create. The authors state that blind people need and want to click photographs for the same reasons as others—recording events, sharing experiences and as a form of artistic expression. There are many BVI people who actively create visual content in the form of blogs covering a wide range of topics, that cater not only to BVI audiences but also to sighted audiences [3]. There is also a thriving community of BVI creators on YouTube [12] and Instagram [2] who offer sighted viewers a glimpse into their lives by sharing their stories and daily experiences, while also being a source of support and guidance to the BVI community.

Assistive technology devices like screen readers, tactile printers, and braille displays have also led to many employment opportunities [10, 18, 25] for BVI people. The number of blind programmers has been steadily rising in the last few years. According to Stack Overflow’s annual Developer Survey, as of 2019, 1.5% of devel-

opers (out of nearly 90,000) reported being BVI [17] as compared to 2017 when 1% of developers (out of 64,000) reported being blind [16]; an increase of nearly 111%.

Although past research in accessible programming tools, such as *CodeTalk* [50] and *StructJumper* [23], has enabled BVI developers to access information better and code more efficiently, they still lack accessible ways to create visual layouts. During my interview with a blind user, he stated that even though he has learnt HTML and CSS, he can only specify *what* element he wants to add but cannot understand or specify *where*. He still has to rely on the assistance of a sighted associate for all aspects of the layout design. Studies have also shown that BVI users are interested in layout design and want to do more than just entering text [41]. Two recent papers [41, 51], that I will discuss more in section 2.3 in the next chapter, address the problem of *editing* visual layouts for BVI user. I will also discuss the Developer Toolkit (DTK) by Andy Borka [6] that helps BVI developers create user interfaces on NVDA screen reader [13] in section 2.3.

The ability to create websites is not just a niche requirement for developers, but a common and useful need for everyone, both from the perspective of employment as well as self-expression. Thus there is a need for accessible web design tools that BVI users can use to create visual layouts. This thesis is an effort to create such a tool by employing a Tangible User Interface (TUI) [33] approach.

1.2 The Tangible Approach

Ability-based design [65] is an approach that advocates shifting the focus of accessible design from disability to ability, much like user-centered design shifted the focus of interactive system design from systems to users. In this approach, designers should strive to leverage all that their target audience *can* do [65]. Blind users use their sense of touch to “navigate and negotiate the world” [65] in their daily life. Previous studies have revealed that blind people have higher tactile acuity than sighted people [28]. Based on this perspective, I see value in a tool that can support this tactile perceptual ability by using a Tangible User Interface [33].

The next chapter will discuss more about TUIs (Section 2.1) and some of their applications in accessibility (Section 2.2). This will help guide the design of the tangible web layout design tool, and also strengthen the case for employing a TUI-based approach for this research.

1.3 Research Questions

As mentioned in the previous sections, there is a need for accessible web design tools that BVI users can use to create visual layouts. This thesis aims to answer the following exploratory questions:

1. What are the roadblocks and difficulties BVI people face when creating visual layouts?
2. How can we effectively translate the visual metaphors of layouts so that that

they can be understood by BVI people?

3. What design opportunities exist to support BVI people to create visual layouts?
4. What are the best approaches for enabling BVI people to create visual layouts?

1.4 Contributions

The contributions of this thesis are as follows:

1. Identification of issues faced by BVI people when creating visual layouts.
2. Highlighting the potential of using a tangible user interface to enable BVI people to design web layouts.
3. Results of a participatory design workshop with a blind co-designer that informs the design of a tangible layout design tool.
4. The design and development of *Sparsha*, a novel *Tangible Layout Design Tool*.
5. An evaluation plan to gauge the effectiveness of the tangible layout design tool.

1.5 Thesis Overview

The chapters of this thesis are structured as follows:

- Chapter 2 describes some of the related research that have laid the foundation for this thesis.

- Chapter 3 describes the Formative Study conducted for this research, and its findings that helped to inform the design of *Sparsha*.
- Chapter 4 lists the design considerations and solutions for *Sparsha*. This is followed by a description of the Participatory Design process and how it was applied in this research to design *Sparsha*.
- Chapter 5 contains the system overview and implementation details of *Sparsha*.
- Chapter 6 presents the plan for testing and evaluating *Sparsha* with BVI participants.
- Chapter 7 wraps it up with limitations, future work, and conclusion.

Chapter 2: Related Work

2.1 Tangible User Interfaces

According to Finnish neurophysiologist Matti Bergström, quoted in a lecture by Sofia Svanteson [5, 19]:

The density of nerve endings in our fingertips is enormous. Their discrimination is almost as good as that of our eyes. If we don't use our fingers during childhood or youth, we become “fingerblind,” this rich network of nerves is impoverished — which represents a huge loss to the brain and thwarts the individual's development as a whole.

Since Ivan Sutherland's Sketchpad [58] 60 years ago, Human-Computer Interaction has largely been in the form of Graphical User Interfaces—a keyboard and mouse being used to manipulate information on a screen by interacting with windows, icons, menus, and pointers (WIMP) [55]. While the quality of the graphics improved, the interfaces themselves have seen little change over the years and for many decades it seemed like there were no alternatives [55]. Around 30 years ago, technology advances and extensive work in HCI research enabled researchers to develop novel forms of post-WIMP interfaces and interactions [55]. Among these

post-WIMP interfaces, Tangible User Interfaces(TUI) were born out of a desire to bridge the gaps between our physical world and the digital world [35].

The earliest introduction of this idea was in *Graspable Interfaces* by Fitzmaurice et al. in 1995 [29]. A few years later Hiroshi Ishii and his student Brygg Ullmer from the Tangible Media Group at MIT Media Lab put forth the idea of *Tangible Bits* [35], which laid the foundation for TUIs. According to Hiroshi Ishii, TUIs let us take advantage of our evolved dexterity in manipulating physical objects with our hands (e.g. building blocks or clay models) to manipulate physical representations of digital information [34].

2.2 TUI Applications and Efficacy

In this section I will explore some of the relevant applications of TUIs and their effectiveness. This will help guide the design approach for implementing the tangible web layout design tool while also strengthening the case for using TUI as a suitable solution.

2.2.1 Learning

Multi-touch interfaces are some of the most widespread post-WIMP interfaces today [8] and provide many of the same benefits as TUIs [62]. In fact they also provide many benefits that TUIs don't [62]. But there are certain scenarios where TUIs outshine multi-touch interfaces. Research suggests that in control tasks, speed of acquisition tasks and the accuracy of manipulation are both higher in TUIs as

compared to multi-touch [62]. Studies and theories on learning have stressed on the positive impact of physical representations, movement and multimodal interactions [22, 48]. TUIs are also very effective when used as an educational tool for young children[53, 67], and especially for those with disabilities [66].

2.2.2 Programming

The origins of tangibles being used for programming predates the very notion of TUIs. Radia Perlman, at Seymour Papert’s lab at MIT, created what is now considered one of the first tangible programming systems [49]. *Tangible Programming Bricks* by Mcnerney was one of the earliest to use electronic blocks to enable kids to code. Lego’s robotic construction kit, *Mindstorms* that was born out of the MIT Media Lab’s Lifelong Kindergarten group [52] and Google’s *Project Bloks* [15] are two more examples that aim to create tangible programming experiences for kids.

Microsoft’s *Project Torino* is a tangible programming language created for BVI children [45, 46]. The authors state that in most of the current programming languages aimed at children, inputs and outputs are visual in nature. As a result, these languages are not accessible for BVI children [46]. Project Torino fills that gap by designing a physical programming language that BVI children (aged 7 to 11) can use to learn coding as well as to develop analytical thinking skills [46].

2.2.3 Accessibility for BVI Users

The ability-based design [65] approach makes the ability of prospective users its central focus. Since BVI people use their fingers and their sense of touch to “navigate and negotiate the world” [65] everyday, it stands to reason that we should take advantage of it when designing solutions for the BVI population. In fact, blind people have been found to have higher tactile acuity than sighted people [28]. Although TUIs have been a focus of HCI research, there is surprisingly little work on TUIs for BVI people [27].

One exception is *The Tangible Desktop* by Baldwin et al. [24] which makes a very strong case for the use of multi-modal interfaces, combining tangible and audio interfaces, in computing systems for BVI users. Their study found that TUIs are quicker and more efficient than the audio-only interfaces that have been, and still are, the main method of computer interaction for BVI computer users for more than four decades [24]. The physical and visual metaphors that make GUI interaction simple for sighted users (e.g. files and folders) do not translate well in a non-visual context, thus making it much more difficult to use than in a visual context [24, 26]. A recent paper *ShapeCAD* by Siu et al. [57] from the Stanford SHAPE Lab describes the use of a TUI for 3D modelling by BVI users. They make use of a table-top, 2.5D shape changing display [56] to enable BVI users to haptically explore and modify existing 3D models, as well as to create new 3D models. I will discuss this some more in section 7.2.

2.2.4 Tactile Graphics

As mentioned earlier, since blind people have been found to have higher tactile acuity than sighted people [28], tactile graphic overlays are a good solution for conveying information to BVI users. *Touchplates* [38] talks about the implementation of a low-cost toolkit to create tactile graphic overlays to support the usage of touchscreens for BVI users. *TacTILE* [30] is a similar work to the previous except that it focuses on making touchscreen mobile phones accessible by using audio-annotated 3D-printed tactile overlays that represent graphical information. Taylor et al. [60] extends the concept of tactile overlays to allow BVI users to interact with customizable 3D printed tactile maps. Holloway et al. [32] goes even beyond tactile maps and proposes that BVI users prefer 3D printed models over traditional tactile maps.

2.3 Visual Layouts for BVI developers

Past research in accessible programming tools for BVI programmers have addressed issues like the inaccessibility of IDEs [43, 50, 54], out-of-context editing [43], and effective code navigation [21, 23]. However, they do not address the accessibility challenges relating to the visual attributes of user interfaces.

Li et al. [41] at the Stanford SHAPE Lab have created a multimodal tool to help BVI people to understand and edit web pages. They used semi-structured interviews and contextual inquiries to identify that the main challenge faced by the BVI populace in understanding and editing layouts is that the screen readers or magnifiers they use fail to communicate enough information about the graphic

design [41]. Their formative research also showed that BVI users are interested in layout design and want to do more than just entering text [41]. Following this, they conducted co-design sessions with a blind, hobbyist web developer and developed a tool that generates tactile print-outs of the layout which users can place on top of a tablet and interact with using a self-voicing application [41].

UITalk by Potluri et al. [51] is an initial exploration of a multi-modal approach to edit web page designs. Their system allows BVI developers to modify visual layouts of web pages by either modifying the code directly in the IDE or by using gestures on a touchscreen. Edits made by either mode are then automatically assessed using a design validator.

Both the above mentioned papers [41, 51] address the editing of existing visual layout templates and not creation of new layouts. Developer Toolkit(DTK) by Andy Borka [6] is an add-on for the open-source NVDA screen reader [13] that helps BVI developers create user interfaces by providing information about the the location, height, and width of web UI elements in pixels.

2.4 Summary

This chapter discusses some of the related works that have laid the foundation upon which this research was built. It begins with a discussion of the origin of tangible user interfaces. Next, it talks about some of the applications of TUIs in relevant application domains. These works helped to strengthen the case for employing a TUI-based approach for this research, and also helped to guide the

design of the tangible web layout design tool. I also discuss some of the works in accessibility for BVI people that employ tactile graphics. Finally I conclude with a discussion of two papers that are addressing the same problem area of visual layouts for BVI users but deal only with editing existing visual layouts and not the creation of new layouts.

Chapter 3: Formative Study

3.1 Introduction

To understand how BVI people use and create websites with currently available tools and technologies, I conducted a formative study consisting of a semi-structured interview [20]. The insights gained from this formative study were used to guide the design of *Sparsha*.

3.2 Interview procedure

I conducted an hour-long, in-person, semi-structured interview [20] with a blind participant to find out how he interacts with websites and about any prior experience with designing or creating any visual layouts like websites or slide decks. I used an [Interview Script](#) with open-ended questions[20] to get more descriptive answers. Data consisted of audio recording and field notes.

3.3 Background

I recruited a blind person who is a student in a Human-Computer Interaction graduate program, as well as a researcher in the field of Accessibility. He agreed

to come on board as a participant for both the interview and as a co-designer for the participatory design session[64] which I will detail in Chapter 4. I recruited him primarily because I needed a BVI user who had experience with the creation of websites using traditional HTML and CSS. Also, being an Accessibility researcher gives him a unique perspective.

The participant had multiple technical certifications and worked as a computer specialist until he lost his eyesight in 2009. He went back to college two years later to get a Bachelor’s degree in Information Studies. Currently, he is in an HCI Master’s program and is also a researcher working on assistive technologies for BVI users.

3.4 Interview Themes

After the interview was conducted, I transcribed the audio recording to get the interview script. Using the process of *coding* [63] I conducted a thematic analysis of the interview data and found the following emergent themes.

3.4.1 Use of assistive technologies

After losing his eyesight, he attended a training center for a few months where he was taught how to navigate life as a blind person and how to use a computer, keyboard navigation, how to use screen readers with Microsoft Word, PowerPoint, Excel and so on. But overall he felt like he did not learn much there and would consider himself largely self-taught. “I can do what they tell me... I can do my coursework. I can do it independently. I can go to the internet, I can find any-

thing”. But because of a lack of systematic training, he still feels he is perhaps at an intermediate level and has more to learn. He does not know braille.

3.4.2 Web development experience

He learnt HTML, CSS, JavaScript, jQuery, and bootstrap as an undergraduate. When asked about his web development experience, he said he has some trouble while using the editors because he doesn’t realize when there are typos or syntax errors. He also faces issues with adding tabs to indent the code because he wouldn’t know how many tabs were there. In these scenarios, he needed a sighted person’s help to help fix the errors. He said that the editors did have some accessibility features but he still struggled with it.

While working on his portfolio website, he said he had used a template which he customized based on what he had learnt in class. He had a sighted scribe assisting him who told him what the colors and the fonts were and if things were in order. He said the template he used was recommended to him by an accessibility researcher as an accessible template.

He has taken a web programming class in his graduate program as well and as his project, he wants to re-do his portfolio website from scratch using everything he learns in class and without the use of a template. He said that if he used a template again he would never develop any usable web skills and wants to have a ”hands-on” experience.

3.4.3 Opinions on learning

The participant is slightly concerned that learning web programming won't really count as a skill since he used a template and had a sighted associate helping. He really wants to be able to code up a website from scratch so that he can finally say he has web design skills. Right now he only has theoretical knowledge of concepts. "...I cannot do anything that is graphical when it comes to CSS... I can, you know, do my HTML, CSS, and put all the colors the way I learnt it. But I don't know what is on the screen".

I gave him a brief description of how the proposed system would ideally be a table-top representation of the screen with some way to mark elements on it to represent element boxes. To this, after a moment's thought, he replied: "...nobody has sat me down and told me what the code looks like on the screen (before this)". He said if someone had explained to him how everything he has typed translates to the screen, he would probably understand HTML and CSS, and their correlation better.

3.5 Key Findings and Design Implications

- Not all BVI people know braille, hence it can't be used as the only source of tactile feedback when designing for the BVI population.
- Participant was formerly sighted, and stated he visualizes concepts in his head.

A congenitally blind participant would possibly have had a different workflow.

Involve a diverse range of BVI participants for the evaluation phase and any subsequent design sessions.

- Participant is largely self-taught and works best when learning on his own, rather than being taught. Hence the tool will have to support him to do that.
- Participant does not want to take help from sighted associates. The tool should enable him to work independently.
- Participant felt that using a template is like a shortcut and he believes that knowing how to create a layout from scratch will mean he has truly learnt the skill of web design. The tool has to support him to create layouts without the use of pre-existing templates.
- Currently he can only specify what element he wants to add but cannot understand or specify where. The tool should allow him to specify a location on the screen where he wants to add the required HTML element.
- His knowledge of web technology is largely theoretical. He has never really been told how changes in code affect what is on the screen. The tool has to give him feedback on the result of his actions.

Chapter 4: Design Considerations and Participatory Design

4.1 Design Criteria and Solutions

Based on the design insights gained from the interview (described in the previous chapter) and the related work, I came up with the following design criteria and solutions for *Sparsha*.

1. **Criteria:** Since BVI users have no way of knowing how their actions translate visually on the screen, a tactile representation of the screen real estate is required so that BVI users can use their fingers to add elements onto it as well as to locate and modify elements that have already been added.

Solution: There would be a base that would be a table-top tactile surface that would be a tangible representation of a computer's screen real-estate.

2. **Criteria:** Graspable tactile pieces are needed to represent HTML elements so that users can easily place them on the required location to add elements or move them from one location to another to edit the size layouts.

Solution: There would be graspable entities, henceforth called beads, that represent html elements. These can be added onto the base at the required locations to specify the corresponding location where the element needs to be

added on the screen.

3. **Criteria:** Users will need an easy way to set the size of the element on the screen.

Solution: Since all HTML elements can be represented as rectangles, sets of four beads will be used to represent an element—one for each corner of the rectangle. To help the user create these rectangles, a constraint mechanism will be needed to constrain the four beads together.

4. **Criteria:** Users need to be able to add different kinds of HTML elements on the TUI.

Solution: The beads will have tactile textures on them that signify the HTML element type they correspond to.

5. **Criteria:** The type, location, and size of the elements placed on the tactile surface need to be translated onto the screen accurately.

Solution: Explore different sensing mechanisms for location detection and bead-type detection (explained further in Section 5.3 in the next chapter). The sensing mechanism should then send the information to a web server which uses it to render the corresponding HTML elements on the screen.

6. **Criteria:** User needs to be given feedback that the action was completed successfully.

Solution: Audio feedback should be enabled on the web app. It announces that the element of specified type has been placed at the specified location.

4.2 Design of Tactile Elements

For the design of the tactile elements (the base, the bead and the constraint mechanism) outlined in the solutions mentioned in the previous section, I conducted a participatory design session with the BVI participant I had interviewed. I 3D printed a few prototypes for each of the elements (described later in [Section 4.5](#)) that would provide a starting point for the design process. The details of the early prototypes, the Participatory Design process, and the results are described in the following sections.

Details of the various approaches considered and the actual implementation of the tool can be found in the next chapter, [Sparsha: A Tangible Layout Design Tool](#).

4.3 What is Participatory Design?

Participatory Design is a design approach where the prospective users of a design participate in the design process as co-designers [\[64\]](#). It can democratize the design process because, instead of the traditional designer-client roles, it makes all the stakeholders collaborate in the design process [\[39\]](#). Research has also shown that, when compared to designers working solo, Participatory Design has a much higher potential for creativity and innovation [\[44, 61\]](#).

4.4 Design Session Goal

The goal of this participatory design session, also known as the co-design session, was to work alongside the participant to design the best solutions for the tactile elements of the tool using the 3D printed artifacts as starting points.

4.5 Artifacts

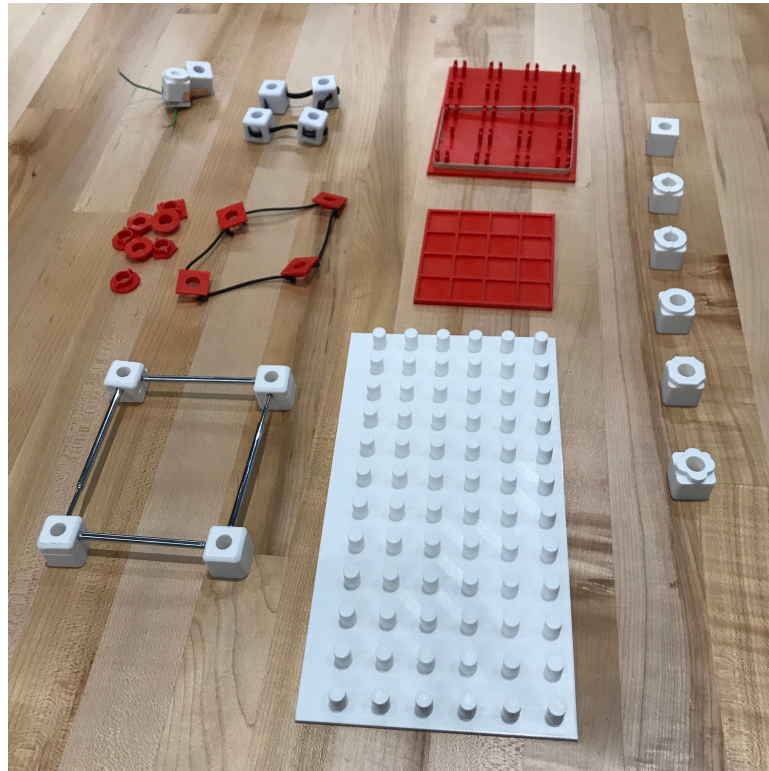


Figure 4.1: The various artifacts used for the co-design session.

Based on the envisioned design solutions listed earlier in this chapter, I started designing and creating the following artifacts for the design session:

1. Designs for a base that would represent the screen real estate were 3D modeled

and printed.

2. Beads that would be used to mark locations on the base. I also created prototypes of multiple types of tactile markers on the beads so that beads can be assigned *content types*. E.g. To add an image element to the layout, I would add four "Image Beads", identified by the tactile markers, on the base in the shape of a rectangle and that would represent an image element on the screen at the same location. The four beads represent the four corners of the rectangle. The beads were 3D modeled and printed as well.
3. A constraint mechanism to mark the edges of the area marked by the beads. This would link the four beads (representing the four corners of a rectangle), help the user align his beads in the shape rectangle on the base to represent a rectangular area occupied by the HTML element on the screen, as well as be a tactile marker when the user is scanning the layout on the base with his fingers.

After iterating over multiple versions of each of the elements, I zeroed in on solution options as the basic artifacts, shown in Figure 4.1 that I would use in the participatory design session.

4.6 Design Session Process

For each tactile element, I presented the user with the different artifacts and let him get a feel for them one by one, followed by a side by side comparison. The



Figure 4.2: User interacting with one of the base artifacts during the co-design session.

user was offered 3 different base options, a board with multiple pillars, a chessboard-like design with raised borders, and a variation of the pillar design with clusters of pillars. Each of the base designs had a corresponding bead design. There were 2 main bead designs—a cuboid design with a hole in the middle that can be placed over a pillared base, and a flatter design with 2 variants that could either be placed on a pillared or the chessboard base. I also tested 3 kinds of constraints used to link the four beads of a set together—using elastic threads, elastic bands, and a telescopic-extension design.

The user was encouraged to think aloud [37, 47] throughout the session and not only talk about the pros and cons of each artifact design but also suggest design changes and improvements that would make the solution more effective. The user was also asked to perform some tasks [40] and his performance was observed (Figure

4.2). The main tasks the user was asked to perform were:

- Find a specified location on the base by feeling the tactile surfaces.
- Place 4 beads in a rectangular shape on the base at a specified location.

4.7 Results and Observations

The user initially said he preferred the flat chess-board design (red) based on the location-finding task, but then switched his preference to the pillared model (Figure 4.3) based on the ease of adding beads to it.

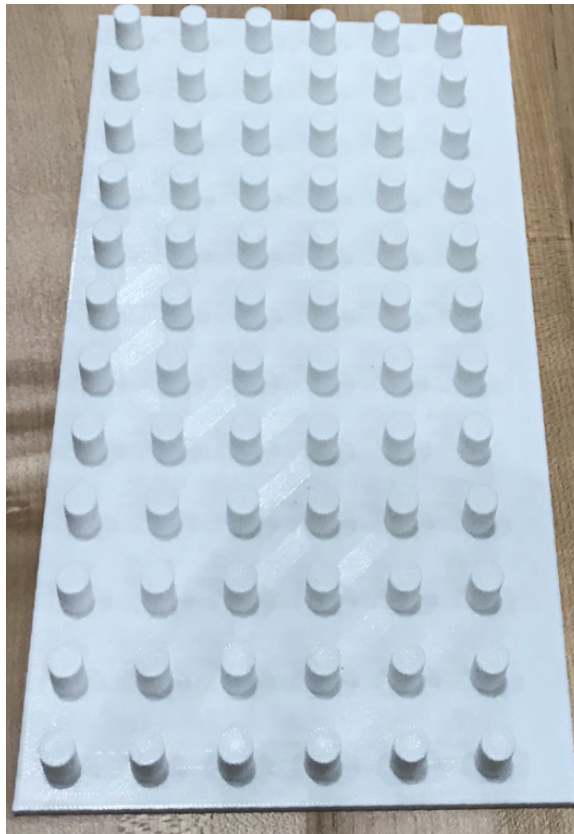


Figure 4.3: The pillared base design

Among the beads, the cuboid design was a no-contest winner over the flat

designs, based on ease of pushing into the base and also to locate on the base later. In the edge constraint designs, while using the elastic string and elastic band models, he struggled a bit in the beginning but eventually made it work. With the telescopic-extension design (Figure 4.4), he took a few moments to realize how it functioned. And then he laughed and said, "Wow this is awesome". He said it was his favorite experience of the session, and said it was very easy for him to create rectangles on the base with it. He did struggle a bit with putting it on the board but that was mainly because of loose joints on the prototype. In all three cases, previously placed beads would pop off when he tried to place the next beads. We discussed that this could be mitigated by adding magnets on both the board and the base.

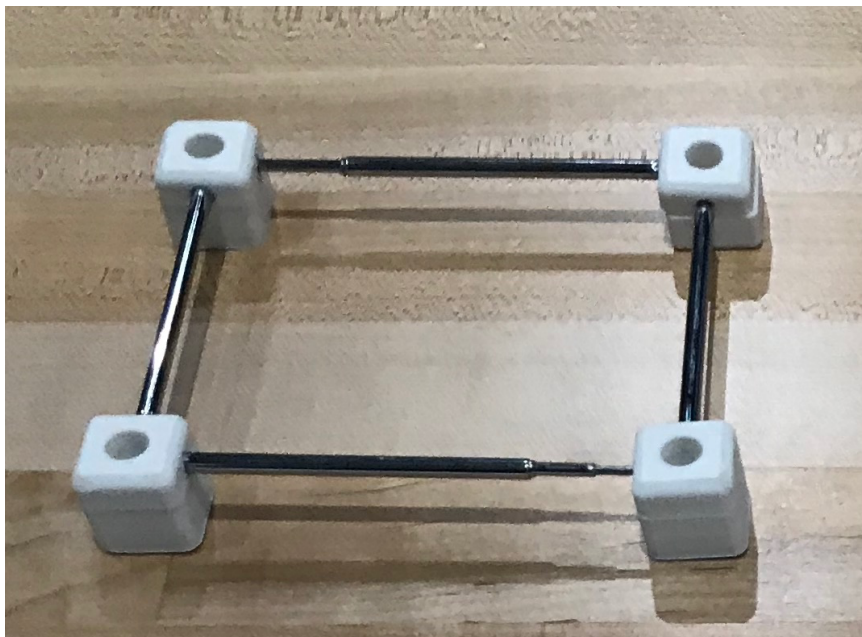


Figure 4.4: The user preferred bead and edge constraint design

Regarding the textures, he was not too keen on the smaller raised markings,

see Figure 4.5 because it was harder to tell the difference between the different markings on different beads. One design had raised markings on the sides of the bead. These could not be sensed with the fingers after the bead was placed on the board and hence the user rejected it. He chose the design with the large shapes on the top 4.5 since they were easier than the others for him to differentiate. He added that the main disadvantage with all the designs would be that he would have to learn which shapes correspond to which element type. An interesting observation was he often couldn't tell the top of beads from the bottom on the beads in Figure 4.4. After trying with other beads, we concluded that it was because that particular prototype had no texture on the top. He said that after texture and magnets were added it would become more obvious.

The most encouraging part of the session was at the end when the user asked if he could have the prototype base, after the project was done, to show his professor and use it in his web programming class. He said, "I finally understand how HTML and CSS work, this will help me a lot."

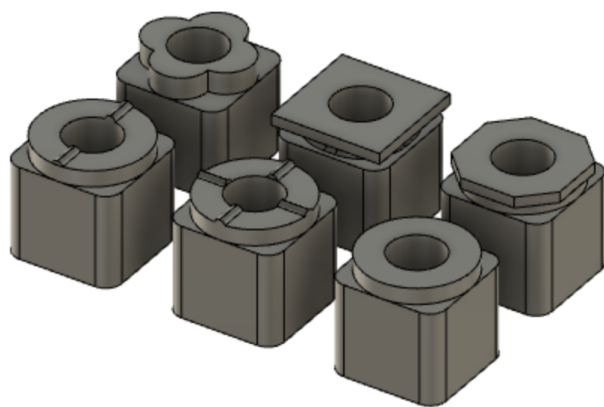


Figure 4.5: The selected bead designs for tactile texture

Chapter 5: Sparsha: A Tangible Layout Design Tool

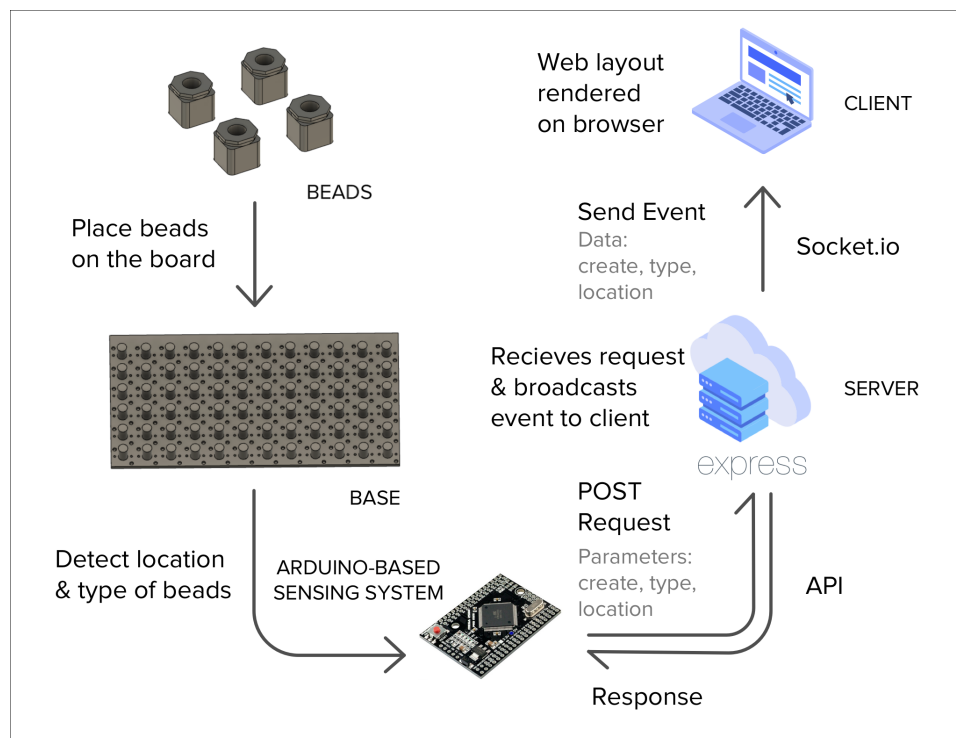


Figure 5.1: Sparsha: System Diagram

5.1 System Overview

The major components of Sparsha can largely be grouped into the electronics, tactile elements, and the web app. The base with its underlying electronics represents the screen real estate, beads act as HTML elements of different types, and

a web application that will get inputs from the hardware and render the HTML layout accordingly.

The system diagram is shown in Figure 5.1. When a user places 4 beads of a type on the board (in the shape of a rectangle due to the edge constraints), they snap on to the board due to magnetic contact points. An Arduino-based sensing circuit attached to the base helps identify the type, and location of the beads. It then sends this information over WiFi to the web server which in turn sends an event to the client browser to render an HTML element of the specified type at the specified location. The web application also gives audio feedback to confirm the placement of the correct HTML element at the correct location.

I chose to go with a 12 column CSS Grid implementation since that is one of the most commonly used and is the most flexible for accommodating different kinds of layouts. Hence, the base has 12 columns of pillars. The number of rows visible in one page(without scrolling) varies according to the row size specified in the CSS grid and also across screen sizes. Based on the row size of 45px that I chose, a grid with 12 rows would have been ideal. Due to the limited size of the 3D printer I decided to reduce the number of rows for this proof of concept to 6.

5.2 Electronics

I initially used an Arduino Uno as the microcontroller to test a scaled-down 3x3 version of the board and later switched to an Arduino Mega 2560 for the full-size board. An ESP8266 Wifi module was used for communicating between the Arduino

and the web app.

5.3 Chess Piece Identification Technology

The two main pieces of information the tool needs to sense are location of the beads on the base, and the type of the bead (to represent different types of HTML elements). Electronic chessboards typically need to find location and type of chess pieces, so I looked at the implementation methods of electronic chessboards [4] for ideas. I explored the different approaches used in electronic chessboards [4]:

- DGT (Digital Game Technology) Sensors [7]: Patented, expensive technology. Hence, ruled out.
- Reed switches [4]: Cheap and reliable. Can detect if a chess piece is in a location, but cannot identify its type. On chess boards, this can be worked around since all the pieces have a fixed starting point but it does not fit my requirement.
- Hall effect sensor [4]: More expensive than reed switches, though still quite cheap. Also they have three pins +, -, data so wiring and code is more complex than reed switches. Probably why they have not been used in commercial systems.
- RFID/NFC [31]: In theory this an ideal solution for chess piece identification. NFC Tags for the beads/pieces are small and cheap. But individual sensors are not cheap and I would need one for each location on the base i.e. 6 x 12

= 72 sensors. In theory, it might be possible to multiplex an RFID antenna. Problems include the difficulty of multiplexing RFID antennas, overlapping reads, and read time.

Another solution was to use a Computer Vision [14] based multiple object tracking algorithm to identify the beads and their position based on visual markers. Downsides for this approach include the need for an overhead camera, algorithm complexity, and the possibility of users occluding the camera view with their hands, which would need additional changes to the algorithm to handle.

Since none of the above approaches seemed ideal for my device, I continued looking for alternatives and I came upon the *Key Switch Matrix*, also known as a Keyboard matrix [11]

5.3.1 Key Switch Matrix

A key switch matrix circuit is commonly found in electronic musical keyboards and computer keyboards and is basically a grid of wires arranged in rows and columns [11]. Each key acts as a switch at every intersecting point of the rows and columns and when a key is pressed, a column makes contact with a row and completes a circuit which is then registered as a key press [11]. The scanning algorithm for the key matrix activates one column at a time and checks the rows for key presses [11]. Multiple key presses with this model cause errors known as *ghosting* and *masking* i.e. registering false key presses and failing to detect when a key isn't pressed anymore [11]. This can be rectified by adding a switching diode in series

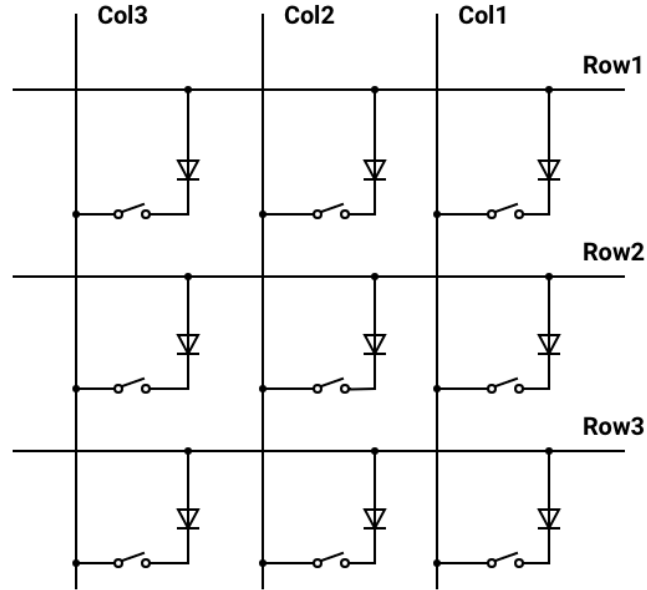


Figure 5.2: 3x3 version of the key switch matrix circuit with diodes.

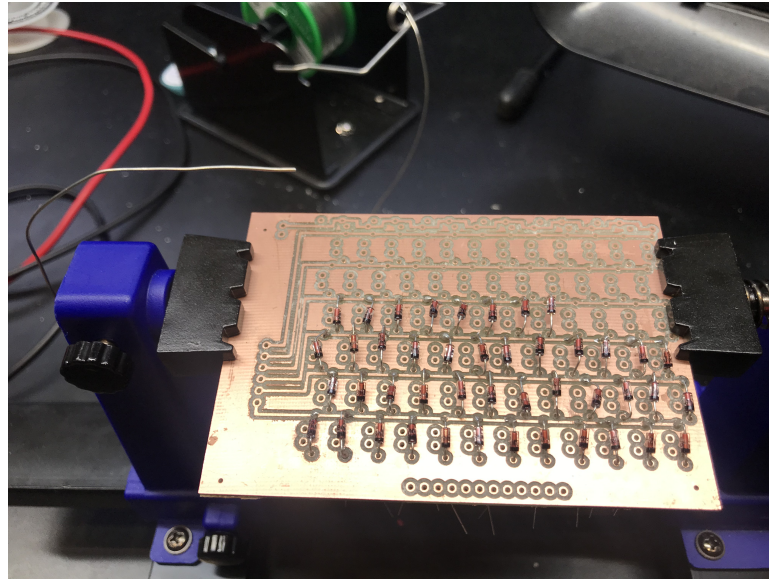


Figure 5.3: Switching Diodes being soldered to the 12x6 PCB

with each switch [11]. In this case, 1N4148 switching diodes were used because of their fast switching times [11]. I also modified the scanning algorithm to activate rows and scan columns instead of the other way around for reasons that will be

explained in 5.3.2. A scaled-down circuit diagram is shown in Figure 5.2.

Adapting the key-matrix to my use case, each pillar on the base would have two connection points acting as the two ends of the switch and when a bead is placed it should close the circuit and detect a "key press" and thus give us the location of the bead.

5.3.2 Matrix of Ohmmeters: Bead Type Recognition

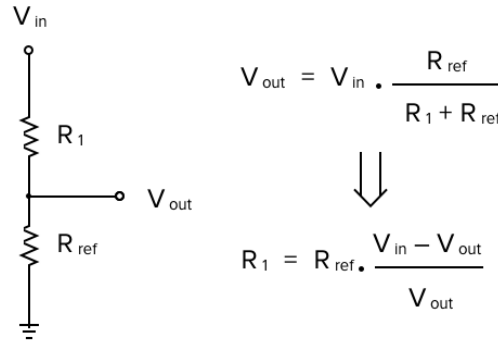


Figure 5.4: Calculating resistance using a voltage divider circuit

Now that location recognition was in place, I had to find a way to differentiate kind of beads. The simplest way was to give each type of bead a specific resistance value and by measuring the resistance the type could be identified. So I essentially turned the matrix of switches into a matrix of ohmmeters. Rather than just detecting a "key press" with an on/off switch mechanism, the device would now detect resistor values at each location on the base. Since measuring the resistance required reading analog values, I switched one side of the key matrix to analog inputs. To account for the limited number of analog pins on the Arduino, I switched the scan-

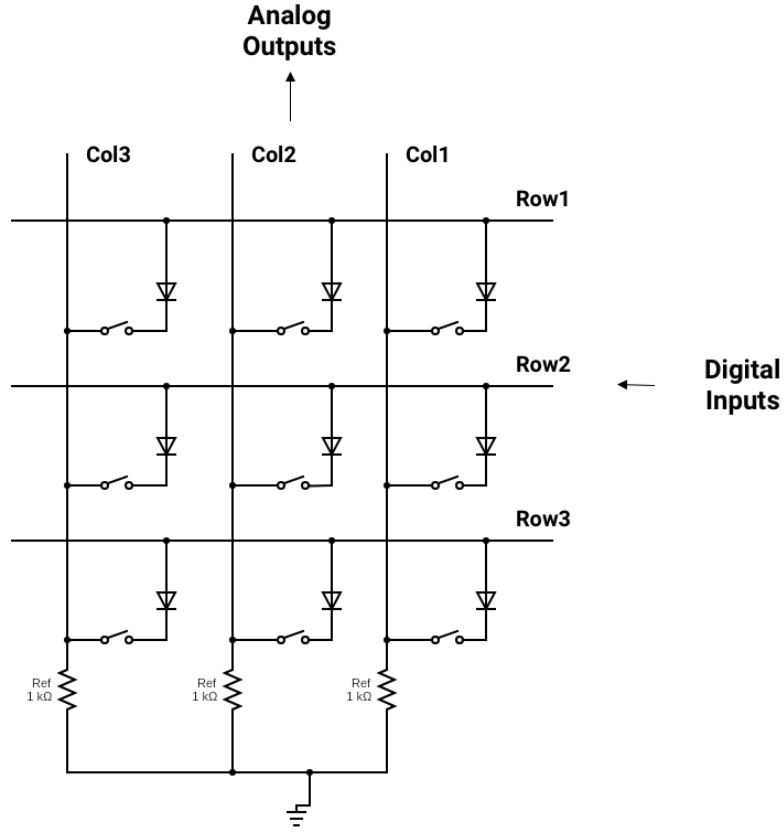


Figure 5.5: Circuit diagram of a 3x3 version with diodes and reference resistors.

ning algorithm to activate rows and scan columns as mentioned above in 5.3.1 and hence fixed the number of analog inputs required to be equal to the number of columns—12.

We can calculate the resistance of each bead using a Voltage divider circuit as shown in Figure 5.4 using a *reference resistor* R_{ref} . I used a reference resistor of 1k ohms which theoretically could be used to measure resistance values between 180 ohms and 5.5k ohms with a measurement accuracy of around 2.5% [42]. The diodes caused some variations to the resistance but I could define ranges of values

to identify various resistors. A scaled-down 3x3 version of the new circuit diagram can be seen in Figure 5.5.

Once the Arduino senses the type and location of the beads, it calls a Create API on the web server (using the Wifi module) passing the type of element, the coordinate of the top-left corner of the rectangle, and the size of the rectangle as parameters.

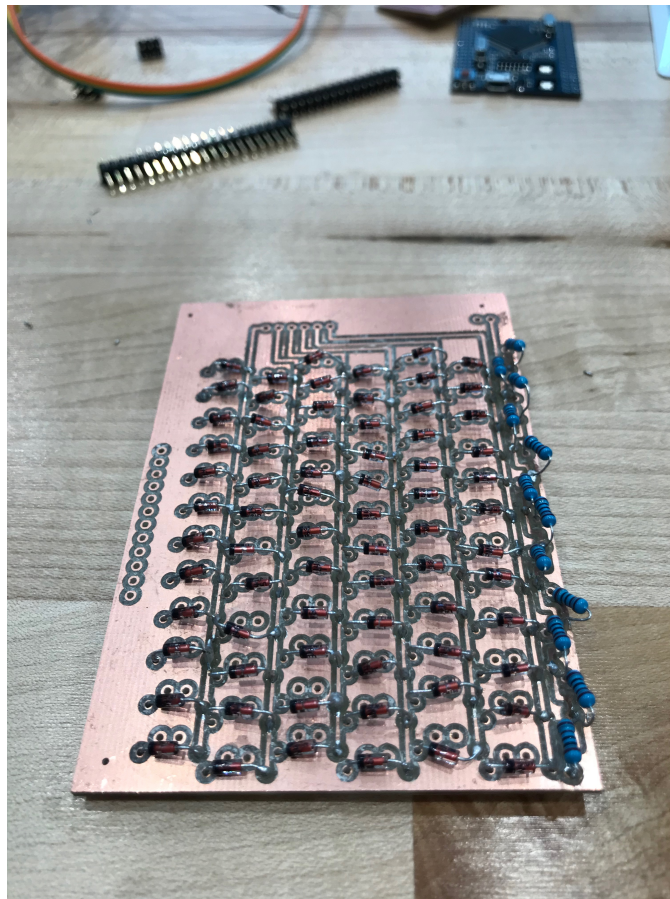


Figure 5.6: PCB after soldering the diodes and resistors

5.4 Tactile elements

5.4.1 Base board



Figure 5.7: The base board design with magnets and copper tape forming the connections

The final base was designed based on the insights gleaned from the co-design session and 3D-printed in a few iterative versions. The iterations helped me to test different positions of connection points, different mechanisms for connection points, and the possibility of using magnets so that beads snap to the pillar and don't get pulled out easily.

The contact points were designed to be diagonal across their respective pillars i.e. one end at the top left of the pillar and the other end on the bottom right. Each



Figure 5.8: 2 pin connectors were used to make connections from the PCB to the base board

contact point would consist of a magnet (to make the bead snap to the location) and adhesive copper tape on top of the magnet so that when the magnets on the bead and base snap together they form a connection. I used JST 1.25mm 2 pin connectors to make the connections from each pillar's contact points on the base to the PCB for ease of removal as seen in Figure 5.8.

5.4.2 Beads and edge constraint design

The bottom of each square bead has 4 magnets covered by copper tape which form 2 diagonal pairs of connection points to ensure that, irrespective of the orientation of the bead, one pair will always make contact with the connection points on the base. The final bead design has the beads hollowed out to contain the resistor

along with tiny holes on the side for the ends of the resistor to make contact with the 2 strips of copper tape which form the connection points. The telescopic-extension edge constraint design which was the preferred choice in the co-design session (as shown in Figure 4.4) was added in the final design. The texture for type identification was added as the shapes-on-top design as was decided in the co-design session (shown in Figure 4.5). One caveat here is that some prior learning will be required for users to know which texture corresponds to a particular HTML element.

5.5 Web Application

The web app consists of a server and a client component. I set up the web server using *Express*, which is a minimal *Node.js* web application framework, and hosted it on Heroku. I set up a *Create* HTTP POST method which gets called by the Arduino.

The API parameters are:

- Type of element
- Column start i.e. column number of the left edge of the rectangle
- Column span i.e. width of the rectangle
- Row start i.e. row number of the top edge of the rectangle
- Row span i.e. height of the rectangle

The server then passes these values to the client-side application via *socket.io*. On the client, I have prepared a 12-column CSS grid container (see Figure 5.9)



Figure 5.9: The 12 column CSS grid container used on the client (if the outlines were visible)

that will be a parent for all the rendered elements. The client then renders an HTML *div* element that contains the HTML element of the specified type and size at the specified location within the grid-container and gives audio feedback that the element was created. At the moment, the layout is populated with dummy content and can be extended to customize the content. Finally, the client provides audio feedback that the element was added at the specified location.

5.6 Bringing it all together

The system diagram is shown in Figure 5.1. When a BVI user places four beads of a particular content type on the board, say of image type, the location and type of the beads are sensed by the Arduino-based sensing circuit. The beads would be in the shape of a rectangle due to the design of our bead constraint. The location of the beads provides the size of the required image element. The Arduino

then calls a Create API using the attached Wifi module and passes the type of the beads (image), the coordinates of the top-left corner of the image, as well as the size of that image as the parameters. The Web server receives this information and sends an event to the client browser to render an image of the given size and at the given location to the client browser. The image element is then rendered on the screen, with a stock image pulled using the Unsplash API [\[1\]](#), along with audio feedback stating that the image element was added at the specified location.

Chapter 6: Summative Evaluation Plan

In this chapter, I present the plan for testing and evaluating the tangible visual layout design tool, *Sparsha*, with BVI participants. First I will describe the participants that need to be recruited, followed by a description of the summative assessment methods to be employed.

6.1 Participants

For the evaluation sessions, I plan to recruit at least 4 to 5 participants across the spectrum of visual impairments so that we have a diverse pool of users. This will provide useful insights about the usability and viability of the layout design tool and inform us about any required additions and modifications to *Sparsha*. Since the co-designer on this project was formerly sighted, it will be useful to even include participants who are congenitally blind and gauge and compare their experiences, behaviour and perspectives. Typically this tool is targeted at BVI people who have background knowledge of technology and web design. But in future work (Section [7.2](#)) it could be interesting to also include participants who are technological novices to test the effectiveness of *Sparsha* as a learning tool.

6.2 Method

The summative evaluation for *Sparsha* will be done in three stages:

1. Screening survey and introduction to *Sparsha*
2. Participants will be asked to perform a given set of tasks, and their performance will be observed and evaluated. A post-study survey will be used to collect the data.
3. The third stage will consist of a semi-structured interview [20] with each of the participants to get their feedback about *Sparsha*.

The next few sub-sections describe the three stages in detail.

6.2.1 Stage 1: Screening and Introduction

In the first stage, participants will be asked to fill in a [Screening survey](#) to give us information about their background, nature of their visual impairment, and details about their technological experience, if any. Following this survey, participants will be given an introduction to *Sparsha*, and will be given a description of how *Sparsha* works and provided some basic guidance on how to use it. Some time will be provided for them to get an initial feel of the system. They will also be given time to learn the feel of various tactile textures on the beads. Participants are encouraged to ask any questions they might have before the tasks begin.

6.2.2 Stage 2: Tasks

Each participant will be asked to perform the following tasks: [40] on Sparsha:

- Identify HTML types from beads based on the texture.
- Add specified HTML element at specified coordinates.
- Add specified HTML elements at vaguely specified locations (eg. Add a photo on the top-left and a paragraph of text to its right).
- Resize HTML elements already on the board.
- Resize elements already on the board.

The participants will be encouraged to think aloud [37, 47] throughout the session and their performance on each task will be observed and evaluated. After each task the participants will be asked to fill in a post-task survey described in the next subsection.

6.2.2.1 Data Collection and Evaluation Metrics

Users will be video recorded during the evaluation. Field notes will be taken to highlight any insightful comments, ideas and suggestions that come up while the user is doing the tasks. The following data pertaining to each task will be collected for each participant in a post-task survey:

- Did the user complete the task?

- Did the user face any issues/roadblocks/errors?
- How much time did the user take to complete the task?
- How satisfied was the user with their performance in the tasks? (on a numbered scale)

Using the above data, *Sparsha* will be evaluated according to the following metrics:

- Success rate: How many users managed to complete the task?
- Error rate: What percentage of users identified the same common errors?
- Time to complete task: What is the average time it took to complete a given task?
- Satisfaction rankings: How satisfied did the users say they were with their performance on the task.

6.2.3 Stage 3: Interview and discussion

The third stage of the summative evaluation will consist of a semi-structured interview [20] with each of the participants about the usability of the web layout design tool as well as their general impressions of the tool. They will be encouraged to share their opinions of the tool, both positive and negative, as well as ideas for improvement. They will also be asked if they have any thoughts and ideas on future feature additions to the tool.

Chapter 7: Discussion and Conclusion

7.1 Limitations

One of the obvious limitations of this tool is the size of the board and the scale of screen real-estate that can be represented. A 12x12 board would be sufficient to represent how much can be seen on a screen at a time. As of now the only solution to design a scrolling page is to split it and design one page at a time, clear the board and start again. Which also highlights the fact that there is no persistence for designs on the tool. It is focused on making the creation of new layouts easy but does not have a way to edit existing layouts.

The current version also does not support overlapping HTML elements because of the way the telescopic-extension beads are built. If it's an important requirement, one option is to use the elastic string design at the cost of reducing ease of use. In the next section, I will talk about the scope of future work where I will also discuss some ways to address these limitations.

7.2 Future Work

Since the tool supports only the creation of new layouts but not editing existing layouts, I would say this cannot be considered an end-to-end web development tool. It does one job, creation, and does it well. I would see it more as a tool that shines when used in conjunction with other feedback-heavy systems such as UITalk [51] or the multimodal tool by Li et al. [41] which are meant for editing existing layouts but cannot create new layouts from scratch. That way we can utilize the strengths of both the tools and mitigate each other’s weaknesses.

Future versions of the tool will benefit greatly if we involve more BVI of various degrees and types of visual impairment and blindness users in the co-design as well as the evaluation stages. For instance, a user who is congenitally blind would have a different perspective than our current user who was formerly sighted. The current tool handles very simple layouts so one avenue for more work is support for advanced HTML tags, CSS features, and content customization. This would put the tool on the path to eventually become a fully-featured accessible Content Management System (CMS).

What I can envision would be the ultimate version of the tool would be this Tangible UI system combined with a shape-changing display such as Stanford SHAPE lab’s Shapeshift [56] to form a kind of refreshable tactile display that also serves as a tangible interface. They have recently implemented ShapeCAD [57] based on a similar premise using the 2.5D display to help BVI users create and edit 3D models. The tool can also be based on a system such as Dynablocks [59]

which can act as a shape-changing display, as well as output graspable 3D shapes for interaction.

7.3 Conclusion

Despite the ever-increasing number of blind and visually impaired programmers, there is still a lack of accessible layout design tools. In this thesis, I present an exploratory attempt at designing and developing a novel, accessible layout design tool, *Sparsha*, that employs a tangible user interface. I interviewed a blind graduate student and conducted a co-design session with him to gain useful insights which formed the basis of the system to be an effective, simple, and low-cost device. I also laid down an evaluation plan for the summative assessment. I believe this project lays a good foundation for future work to eventually build a tangible web design and CMS tool.

During the discussion that I had with the user at the end of the codesign session, he said he was already able to get a better sense of how his HTML code gets manifested on the screen by playing around with the lo-fi prototypes. The fact that he asked to borrow the prototype for his coding class shows that this approach shows promise and can be a very effective learning tool for BVI users. The user mentioned that he could even imagine his young daughters using such a system to learn to create web layouts.

Appendix A: Interview Script

1. Can you talk about your background?
2. Can you tell me about your history with technology? Could be in coding, design, as a digital consumer, slide decks or any other way.
3. Can you describe and demonstrate how you traverse and read information from websites?
4. Do you have experience with making websites? If yes, can you talk about it?
5. Can you talk about the accessibility features that make your work in coding/design/tech easier? (Based on answer to Q2)
6. Can you tell me about how the lack of certain accessibility features has affected the work you do in coding/design/tech? (Based on answer to Q2)
7. Follow-up to Q6: What feature/ product do you wish existed that could improve, elevate or even enable you to do certain work in coding/design/tech. (Based on answer to Q2)
8. How important is it for you to be able to make your own websites? Has your inability to make a website hindered you in anyway?

9. What, do you imagine, would be an ideal solution for you to make websites
would be like?
10. Would you be interested in participating in co-design sessions?

Appendix B: Screening survey

1. What is your age?
2. What kind of visual impairment do you have?
3. Are you interested in participating in this study?
4. Why do you want to participate in this study?
5. Do you use technology on a regular basis?
6. Are you a student/professional in the technology field?
7. Do you have any experience with coding/app/web design or creating slide decks?
8. Can we contact you with follow-up questions after the task?
9. Do you have any additional questions or comments?

Bibliography

- [1] Beautiful Free Images & Pictures — Unsplash, . URL <https://unsplash.com/>. (Last accessed on 04/01/2020).
- [2] Beauty influencers with disabilities you need to follow right now, . URL <https://thetempest.co/2018/12/03/style/beauty-influencers-disability/>. (Last accessed on 04/01/2020).
- [3] Blogging when you're blind or visually impaired. URL <https://yourdolphin.com/news?id=223>. (Last accessed on 03/03/2020).
- [4] Chess Piece Identification Technology - Arduino & Raspberry PI Chess Computer. URL <http://chess.forththerapy.co.uk/home/chess-piece-identification-technology/>. (Last accessed on 04/01/2020).
- [5] Designing For The Tactile Experience — Smashing Magazine. URL <https://www.smashingmagazine.com/2018/04/designing-tactile-experience/>. (Last accessed on 04/04/2020).
- [6] Developer Toolkit. URL <https://addons.nvda-project.org/addons/developerToolkit.uk.html>. (Last accessed on 04/06/2020).
- [7] Electronic Boards & - Digital Game Technology. URL <http://www.digitalgametechnology.com/index.php/products/electronic-boards>. (Last accessed on 04/01/2020).
- [8] How Many People Have Smartphones Worldwide (Apr 2020), . URL <https://www.bankmycell.com/blog/how-many-phones-are-in-the-world>. (Last accessed on 04/04/2020).
- [9] How accessible is social media if you have a visual impairment? - Life of a Blind Girl, . URL <https://lifeofablindgirl.com/2019/04/03/how-accessible-is-social-media-if-you-have-a-visual-impairment/>. (Last accessed on 04/01/2020).
- [10] How Technology Brings Blind People into the Workplace, . URL <https://hbr.org/1989/03/how-technology-brings-blind-people-into-the-workplace>. (Last accessed on 03/09/2020).

- [11] Keyboard Matrix Help. URL https://www.dribin.org/dave/keyboard/one_html/. (Last accessed on 04/04/2020).
- [12] Meet the Blind YouTubers Making the Internet More Accessible — WIRED. URL <https://www.wired.com/story/blind-youtube-creators/>. (Last accessed on 03/26/2020).
- [13] NV Access. URL <https://www.nvaccess.org/>. (Last accessed on 04/06/2020).
- [14] OpenCV. URL <https://opencv.org/>. (Last accessed on 04/01/2020).
- [15] Project Bloks - Research. URL <https://projectbloks.withgoogle.com/research/>. (Last accessed on 03/31/2020).
- [16] Stack Overflow Developer Survey 2017, . URL <https://insights.stackoverflow.com/survey/2017>. (Last accessed on 03/09/2020).
- [17] Stack Overflow Developer Survey 2019, . URL <https://insights.stackoverflow.com/survey/2019#demographics>. (Last accessed on 03/30/2020).
- [18] The Tools of a Blind Programmer—Parham Doustdar’s Blog, . URL <https://www.parhamdoustdar.com/2016/04/03/tools-of-blind-programmer/>. (Last accessed on 03/09/2020).
- [19] The Future of interaction design 1 - Sofia Svanteson - YouTube, . URL https://www.youtube.com/watch?v=J_dXLjx-NcY. (Last accessed on 04/04/2020).
- [20] William Adams. Conducting Semi-Structured Interviews. 4 2015. doi: 10.1002/9781119171386.ch19.
- [21] Khaled Albusays, Stephanie Ludi, and Matt Huenerfauth. Interviews and Observation of Blind Software Developers at Work to Understand Code Navigation Challenges. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS ’17, pages 91–100, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349260. doi: 10.1145/3132525.3132550. URL <https://doi.org/10.1145/3132525.3132550>.
- [22] Daniel Avrahami and Scott Hudson. Forming Interactivity: A Tool for Rapid Prototyping of Physical Interactive. pages 141–146, 4 2002. doi: 10.1145/778712.778735.
- [23] Catherine M Baker, Lauren R Milne, and Richard E Ladner. StructJumper: A Tool to Help Blind Programmers Navigate and Understand the Structure of Code. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI ’15, pages 3043–3052, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450331456. doi: 10.1145/2702123.2702589. URL <https://doi.org/10.1145/2702123.2702589>.

- [24] Mark Baldwin, Gillian Hayes, Oliver Haimson, Jennifer Mankoff, and Scott Hudson. The Tangible Desktop: A Multimodal Approach to Nonvisual Computing. *ACM Transactions on Accessible Computing*, 10:1–28, 8 2017. doi: 10.1145/3075222.
- [25] Edward Bell and Natalia Mino. Employment Outcomes for Blind and Visually Impaired Adults. *Journal of Blindness Innovation and Research*, 5, 1 2015. doi: 10.5241/5-85.
- [26] Yevgen Borodin, Jeffrey P Bigham, Glenn Dausch, and I V Ramakrishnan. 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)*, W4A '10, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781450300452. doi: 10.1145/1805986.1806005. URL <https://doi.org/10.1145/1805986.1806005>.
- [27] Anke Brock. Tangible Interaction for Visually Impaired People: why and how. 4 2017.
- [28] Zaira Cattaneo and Tomaso Vecchi. *Blind Vision: The Neuroscience of Visual Impairment*. 4 2011.
- [29] George W Fitzmaurice, Hiroshi Ishii, and William A S Buxton. Bricks: Laying the Foundations for Graspable User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, pages 442–449, USA, 1995. ACM Press/Addison-Wesley Publishing Co. ISBN 0201847051. doi: 10.1145/223904.223964. URL <https://doi.org/10.1145/223904.223964>.
- [30] Liang He, Zijian Wan, Leah Findlater, and Jon Froehlich. TacTILE: A Preliminary Toolchain for Creating Accessible Graphics with 3D-Printed Overlays and Auditory Annotations. pages 397–398, 4 2017. doi: 10.1145/3132525.3134818.
- [31] Steve Hinske and Marc Langheinrich. An rfid-based infrastructure for automatically determining the position and orientation of game objects in tabletop games1. 2007.
- [32] Leona Holloway, Kim Marriott, and Matthew Butler. Accessible Maps for the Blind: Comparing 3D Printed Models with Tactile Graphics. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356206. doi: 10.1145/3173574.3173772. URL <https://doi.org/10.1145/3173574.3173772>.
- [33] Hiroshi Ishii. *Tangible bits: Beyond pixels*. 1 2008. doi: 10.1145/1347390.1347392.
- [34] Hiroshi Ishii. The Tangible User Interface and Its Evolution. *Commun. ACM*, 51(6):32–36, 6 2008. ISSN 0001-0782. doi: 10.1145/1349026.1349034. URL <https://doi.org/10.1145/1349026.1349034>.

- [35] Hiroshi Ishii and Brygg Ullmer. Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, CHI '97, pages 234–241, New York, NY, USA, 1997. Association for Computing Machinery. ISBN 0897918029. doi: 10.1145/258549.258715. URL <https://doi.org/10.1145/258549.258715>.
- [36] Chandrika Jayant, Hanjie Ji, Samuel White, and Jeffrey Bigham. Supporting Blind Photography. *ASSETS'11: Proceedings of the 13th International ACM SIGACCESS Conference on Computers and Accessibility*, 3 2011. doi: 10.1145/2049536.2049573.
- [37] Anker Helms Jørgensen. Thinking-aloud in user interface design: A method promoting cognitive ergonomics. *Ergonomics*, 33, 4 1990. doi: 10.1080/00140139008927157.
- [38] Shaun Kane, Meredith Morris, and Jacob Wobbrock. Touchplates: Low-Cost Tactile Overlays for Visually Impaired Touch Screen Users. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility, ASSETS 2013*, 4 2013. doi: 10.1145/2513383.2513442.
- [39] María Laitano. Developing a Participatory Approach to Accessible Design. *International Journal of Sociotechnology and Knowledge Development*, 9:1–11, 4 2017. doi: 10.4018/IJSKD.2017100101.
- [40] C Lewis and J Rieman. Task Centered User Interface Design: A Practical Introduction. 4 1993.
- [41] Jingyi Li, Son Kim, Joshua Miele, Maneesh Agrawala, and Sean Follmer. *Editing Spatial Layouts through Tactile Templates for People with Visual Impairments*. 4 2019. ISBN 978-1-4503-5970-2. doi: 10.1145/3290605.3300436.
- [42] Gergely Makan, Robert Mingesz, and Zoltán Gingl. How accurate is an Arduino Ohmmeter? *Physics Education*, 54:33001, 4 2019. doi: 10.1088/1361-6552/ab0910.
- [43] Sean Mealin and Emerson Murphy-Hill. *An exploratory study of blind software developers*. 9 2012. ISBN 978-1-4673-0852-6. doi: 10.1109/VLHCC.2012.6344485.
- [44] Val Mitchell, Tracy Ross, Andrew May, Ruth Sims, and Christopher J Parker. Empirical investigation of the impact of using co-design methods when generating proposals for sustainable travel solutions. 4 2015. URL https://repository.lboro.ac.uk/articles/Empirical_investigation_of_the_impact_of_using_co-design_methods_when_generating_proposals_for_sustainable_travel_solutions/9348323.
- [45] Cecily Morrison, Nicolas Villar, Alex Hadwen-Bennet, Tim Regan, Daniel Cletheroe, Anja Thieme, and Sue Sentance. Physical Programming for

- Blind and Low Vision Children at Scale. *Human-Computer Interaction*, 7 2019. URL <https://www.microsoft.com/en-us/research/publication/physical-programming-for-blind-and-low-vision-children-at-scale/>.
- [46] Cecily Morrison, Nicolas Villar, Anja Thieme, Oscar Salandin, Daniel Cletheroe, Greg Saul, Darren Edge, Martin Grayson, and Haiyan Zhang. Torino: A tangible programming language inclusive of children with visual disabilities. *Human-Computer Interaction*, 35(3), 1 2019. URL <https://www.microsoft.com/en-us/research/publication/torino-a-tangible-programming-language-inclusive-of-children-with-visual-disabilities/>.
 - [47] Erica Olmsted-Hawala, Elizabeth Murphy, Sam Hawala, and Kathleen Ashenfelter. Think-aloud protocols: A comparison of three think-aloud protocols for use in testing data-dissemination web sites for usability. In *Conference on Human Factors in Computing Systems - Proceedings*, volume 4, pages 2381–2390, 4 2010. doi: 10.1145/1753326.1753685.
 - [48] Claire O’Malley and Danae Fraser. Literature Review in Learning with Tangible Technologies. *NESTA Futurelab Rep.*, 12, 4 2004.
 - [49] Radia Perlman. TORTIS: Toddler’s Own Recursive Turgle Interpreter System. 10 2004.
 - [50] Venkatesh Potluri, Priyan Vaithilingam, Suresh Iyengar, Y Vidya, Manohar Swaminathan, and Gopal Srinivasa. CodeTalk: Improving Programming Environment Accessibility for Visually Impaired Developers. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI ’18, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356206. doi: 10.1145/3173574.3174192. URL <https://doi.org/10.1145/3173574.3174192>.
 - [51] Venkatesh Potluri, Liang He, Christine Chen, Jon Froehlich, and Jennifer Mankoff. *A Multi-Modal Approach for Blind and Visually Impaired Developers to Edit Webpage Designs*. 10 2019. ISBN 978-1-4503-6676-2. doi: 10.1145/3308561.3354626.
 - [52] Mitchel Resnick. Behavior Construction Kits. *Commun. ACM*, 36(7):64–71, 7 1993. ISSN 0001-0782. doi: 10.1145/159544.159593. URL <https://doi.org/10.1145/159544.159593>.
 - [53] Mitchel Resnick, Fred Martin, Robert Berg, Rick Borovoy, Vanessa Colella, Kwin Kramer, and Brian Silverman. Digital Manipulatives: New Toys to Think With. 4 1970. doi: 10.1145/274644.274684.
 - [54] Emmanuel Schanzer, Sina Bahram, and Shriram Krishnamurthi. Accessible AST-Based Programming for Visually-Impaired Programmers. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*,

- SIGCSE '19, pages 773–779, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450358903. doi: 10.1145/3287324.3287499. URL <https://doi.org/10.1145/3287324.3287499>.
- [55] Orit Shaer and Eva Hornecker. Tangible User Interfaces: Past, Present, and Future Directions. *Foundations and Trends in Human-Computer Interaction*, 3:1–137, 4 2009. doi: 10.1561/11000000026.
 - [56] Alexa Siu, Eric Gonzalez, Shenli Yuan, Jason Ginsberg, Allen Zhao, and Sean Follmer. shapeShift: A Mobile Tabletop Shape Display for Tangible and Haptic Interaction. pages 77–79, 4 2017. doi: 10.1145/3131785.3131792.
 - [57] Alexa Siu, Son Kim, Joshua Miele, and Sean Follmer. shapeCAD: An Accessible 3D Modelling Workflow for the Blind and Visually-Impaired Via 2.5D Shape Displays. pages 342–354, 3 2019. ISBN 978-1-4503-6676-2. doi: 10.1145/3308561.3353782.
 - [58] Ivan E Sutherland. Sketch Pad a Man-Machine Graphical Communication System. In *Proceedings of the SHARE Design Automation Workshop*, DAC '64, page 6.329–6.346, New York, NY, USA, 1964. Association for Computing Machinery. ISBN 9781450379328. doi: 10.1145/800265.810742. URL <https://doi.org/10.1145/800265.810742>.
 - [59] Ryo Suzuki, Junichi Yamaoka, Daniel Leithinger, Tom Yeh, Mark Gross, Yoshihiro Kawahara, and Yasuaki Kakehi. Dynablock: Dynamic 3D Printing for Instant and Reconstructable Shape Formation. pages 99–111, 4 2018. doi: 10.1145/3242587.3242659.
 - [60] Brandon Taylor, Anind Dey, Dan Siewiorek, and Asim Smailagic. Customizable 3D Printed Tactile Maps as Interactive Overlays. pages 71–79, 4 2016. doi: 10.1145/2982142.2982167.
 - [61] Jakob Trischler, Simon J Pervan, Stephen J Kelly, and Don R Scott. The Value of Codesign: The Effect of Customer Involvement in Service Design Teams. *Journal of Service Research*, 21(1):75–100, 2018. doi: 10.1177/1094670517714060. URL <https://doi.org/10.1177/1094670517714060>.
 - [62] Philip Tuddenham, David Kirk, and Shahram Izadi. Graspables Revisited: Multi-Touch vs. Tangible Input for Tabletop Displays in Acquisition and Manipulation Tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 2223–2232, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781605589299. doi: 10.1145/1753326.1753662. URL <https://doi.org/10.1145/1753326.1753662>.
 - [63] Porcia Vaughn and Cherie Turner. Decoding via Coding: Analyzing Qualitative Text Data Through Thematic Coding and Survey Methodologies. *Journal of Library Administration*, 56(1):41–51, 2016. doi: 10.1080/01930826.2015.1105035. URL <https://doi.org/10.1080/01930826.2015.1105035>.

- [64] Maja Velden and Christina Mörtberg. Participatory Design and Design for Values. In *Handbook of Ethics, Values, and Technological Design: Sources, Theory, Values and Application Domains*, pages 1–22. 4 2014. ISBN 978-94-007-6994-6. doi: 10.1007/978-94-007-6994-6{_}33-1.
- [65] Jacob O Wobbrock, Shaun K Kane, Krzysztof Z Gajos, Susumu Harada, and Jon Froehlich. Ability-Based Design: Concept, Principles and Examples. *ACM Trans. Access. Comput.*, 3(3), 4 2011. ISSN 1936-7228. doi: 10.1145/1952383.1952384. URL <https://doi.org/10.1145/1952383.1952384>.
- [66] Matej Zajc and Andreja Istenic Starcic. *Potentials of the Tangible User Interface (TUI) in Enhancing Inclusion of People with Special Needs in the ICT-Assisted Learning and e-Accessibility*, volume 7327. 6 2012. doi: 10.1007/978-3-642-30947-2{_}30.
- [67] Oren Zuckerman, Saeed Arida, and Mitchel Resnick. Extending Tangible Interfaces for Education: Digital Montessori-Inspired Manipulatives. In *CHI 2005: Technology, Safety, Community: Conference Proceedings - Conference on Human Factors in Computing Systems*, 4 2005. doi: 10.1145/1054972.1055093.