THE UNIVERSITY OF CHICAGO

NETWORKBOTS: SWARM USER INTERFACES AS A PLATFORM FOR PHYSICALIZING NETWORK-BASED DATA

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE PHYSICAL SCIENCES
IN CANDIDACY FOR THE DEGREE OF
BACHELOR OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

BY
RAMARKO BHATTACHARYA

CHICAGO, ILLINOIS
APRIL 2024

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

Thank you to Ken for serving as an incredible mentor over the last two years of my research career. You changed how I see technology and opened the door to life-changing opportunities and a community of researchers, designers, and creators I couldn't have imagined. From Austria to Ireland to Austin, you've helped me explore both the world and the limits of technology. You changed how I imagine the future and let me be a part of it.

Thank you, Professor Kindlmann, for deciding to help so much with this thesis. You went above and beyond as a reader, giving me advice on how that helped improve this thesis and will help improve all of my future work.

None of the work presented here would have been possible without all of my friends at the Actuated Experience lab. From campfires at the point to late nights at the Danube, this lab has provided me with two years of incredible experiences that have changed me. First and foremost, my Baxtie, Lillith Yu. We joined together two years ago, and I never could've imagined how close we would become. From late nights at the lab to early mornings at Ars Electronica, the experiences we shared have guaranteed that you will always be my best friend at this lab. To Jonathon Lindstrom, my partner in crime. We worked on a project that seemed impossible for weeks and then became a reality in a matter of days. Your humor and intelligence made you one of my closest friends in only a matter of days, and our work together will stand the test of time. To Emilie Faracci, Harrison Dong, and You Li, we turned Threading Space from drawing into an exhibition together. Through trials and tribulations, we turned a simple idea into an incredible experience. Not just for those who saw the final product but also for us, who made it a reality. Finally, thank you to Raymond, who joined me on the journey of turning NetworkBots into something tangible. Along the way, you helped the project evolve over the course of exhibitions and presentations.

To my AxLab PhD Mentors, Willa Yang, Anup Sathya, and Ran Zhou. Thank you for the guidance and help you three have given me. Despite your experiences, you always treated me as an equal and let me learn alongside you. Willa, time and time again, you have proven to be a great friend and a close confidant. From rock climbing to late-night talks in the lab, you've always cheered my mood and helped me become a more well-rounded researcher. Anup, we had the rare opportunity to make a class together, and even though we helped teach it, I never stopped learning from you. Ran, thank you so much for the advice you have given me over the few months I've known you. Even in that time, you've helped me improve so much as a designer and a researcher.

Thank you to Antares for planting the seeds of this project and watching it transform from a simple idea to something real. This would not been possible without your guidance, feedback, and help.

Finally, thank you to my family. Without the way you fostered my love for technology and science, I never would have been here or searching for the questions I am now.

ABSTRACT

Data Physicalization is an emerging area of data representation using a wide variety of physical data representations. Through powerful tools such as physical artifacts and tangible and actuated user interfaces, data can be presented to users so that they can perceive and interact with digital information in the real world. This thesis introduces NetworkBots, a novel approach that integrates Swarm User Interfaces (SwarmUIs) with Network Visualizations to create an interactive swarm-based network physicalization. NetworkBots facilitates the physical representation of networks, empowering users to comprehend and apply various network-related methodologies and tasks tangibly. This thesis also discusses Laptop-Toio, a platform for prototyping tangible user interfaces using mobile wheeled robots. This work delves into the interaction space, implementation details, potential applications, inherent limitations, and avenues for future research within the realm of NetworkBots.

CHAPTER 1

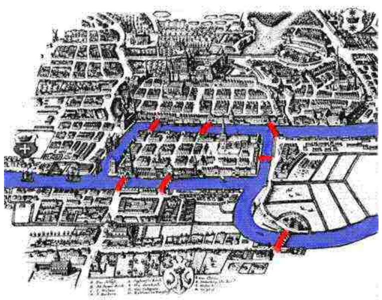
INTRODUCTION

The true delight is in the finding out rather than in the knowing.

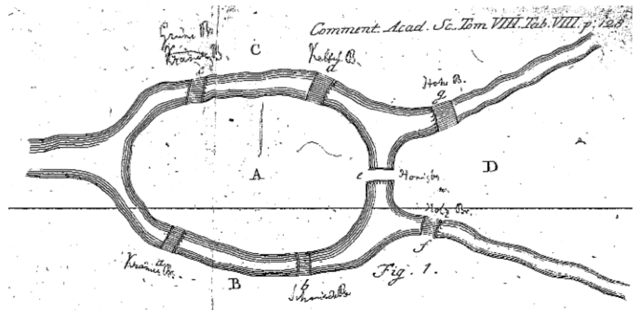
— Issac Asimov

In 1736, a resident of the Prussian city of Königsberg reached out to the mathematician Euler with a problem that had been troubling the local populace: finding a path that would cross all of the city's seven bridges exactly once (Figure 1.1). While Euler initially dismissed the problem as trivial, he quickly found himself immersed. He wrote, "This question is so banal, but seemed to me worthy of attention in that [neither] geometry, nor algebra, nor even the art of counting was sufficient to solve it." Eventually, he found his solution: such a path could not exist. In the process of solving this problem, Euler pioneered an entirely new field of mathematics: Graph Theory [2, 3, 4]. Today, graphs, also known as networks, are used to represent a wide variety of information, from genomic data to the internet. They have been incredibly impactful in the development of computing, as well as a valuable tool for understanding abstract information about the world around us. Our understanding of networks only exists due to Euler approaching what seemed like a simple problem in a novel way. Reflecting on the problem of Königsberg's seven bridges, one can't help but ask: How does our understanding of the world influence our approach to solving problems?

Humanity has been attempting to comprehend the world around us for millennia. Even before computers, people have been constructing *visualizations*, which express information through visuals, and *physicalizations*, which express information through physical artifacts. The oldest known visualizations are cave paintings in Spain, dating back over 64,000 years [6]. The use of physical objects as accounting tokens in Mesopotamia occurred over 4,000 years ago [7]. For millenia, people have been carrying on the grand tradition of visualizing and physicalizing information. The first known use of visualization to represent data was the Turin Papyrus map (Figure 1.2), which recorded mineral distribution and mining data



(a) The seven bridges of Königsberg [5]



(b) Euler's visualization of the seven bridges problem [2]

Figure 1.1: The Puzzle of Königsberg's Seven Bridges

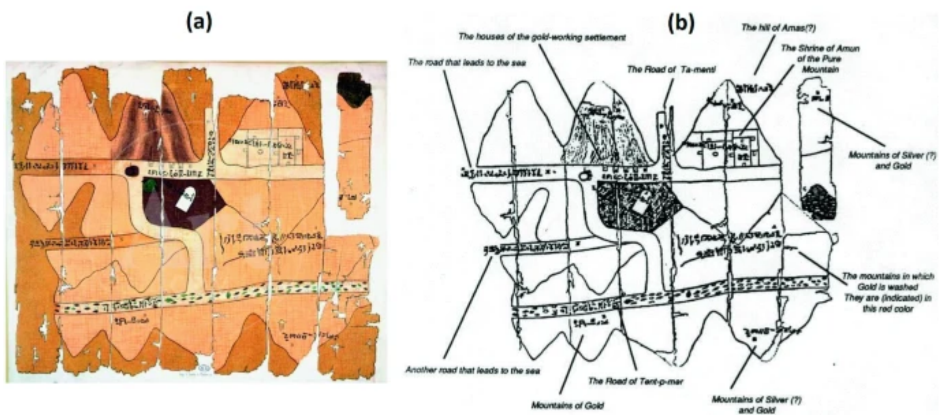


Figure 1.2: The Turin Papyrus map (1150 BC) (a), as well as an interpretation of the map (b) [1]

dating back to 1150 BC [8]. In 1644, Michael Florent Van Langren is thought to have created the first visualization of statistical data [9, 8].

In recent times, pioneers like Tufte, Cleveland, and McGill defined guiding principles and systems for Data Visualization [9, 10]. These works provided a clear outline for how data visualizations can use channels of information like position, size, and color to present information to users. When appropriately used, visualizations can act as a powerful tool for communicating information. Jansen et al. recognized how the growing alternative of Data Physicalizations used physical forms to express data, analogous to data visualizations [11]. In particular, she described how the developing tools of fabrication, tangible user interfaces, and shape displays were serving as new platforms to express data. One example of this

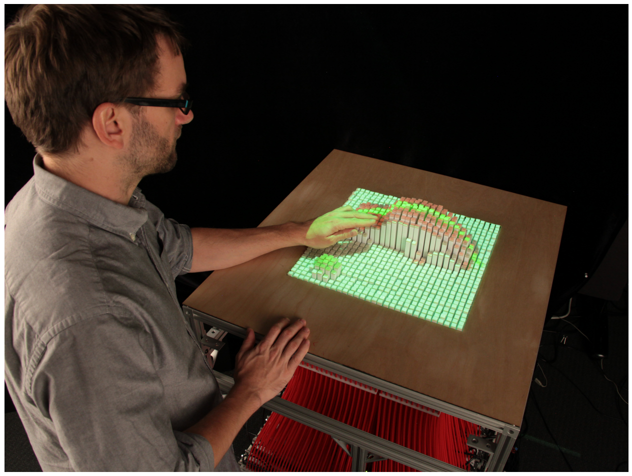


Figure 1.3: Using inForm, a tangible user interface to display geospatial information

growing field was inForm (Figure 1.3), which used a tangible interface of moving pins to present geospatial and other topological information to users [12].

One unique form of tangible computing has arisen in the form of Swarm User Interfaces (SwarmUIs) [13]. Whereas more traditional tangible user interfaces rely on a singular interactive object, SwarmUIs use multiple independent objects as a collective interface. This allows for novel interactions that can only occur with a collective. While SwarmUIs provide a wide range of potential interactions, they also offer new advantages in Data Physicalization, as researchers have explored how SwarmUIs can represent multiple data points in a single physicalization [14]. However, to date, no one has explored how SwarmUIs can be used to express network-based information. Much like the bridges of Königsberg, NetworkBots approaches the familiar challenge of representing data with the new approach of physicalizing networks.

NetworkBots fuses Swarm User Interfaces with Network Visualizations to construct an interactive swarm-based data physicalization. NetworkBots builds on existing research exploring how information can be analyzed using visuals and haptics in tangible user interfaces. NetworkBots allows for the physicalization of networks and network algorithms, allowing users to tangibly understand and apply several network-based methodologies, such as rearrangement, clustering, and searching. etc. In order to accomplish this, this thesis also discusses the development of Laptop-Toio, a toolkit I developed to prototype tangible user interfaces using commercially available mobile robots. I also discuss the possible interactions and applications of NetworkBots, as well as its current limitations. NetworkBots, as well as its interaction space, can be generalized to several fields and disciplines, such as internet information, chemistry, and biology.

In summary, the contributions of this thesis are:

- A general approach and design space for NetworkBots to allow people to interact with a tangible Network Visualization that builds on existing work on data physicalizations and swarm user interfaces.
- Laptop-Toio, an open-source platform for controlling toios, commercially available mobile robots that serve as a popular prototyping tool in Human-Computer Interaction.
- An implementation of a proof-of-concept prototype of NetworkBots, as well as a demonstration of potential applications of NetworkBots.
- An evaluation and discussion of NetworkBots and Laptop-Toio on a technical and design level, discussing its strengths, limitations, and possible improvements.

CHAPTER 2

RELATED WORKS

NetworkBots builds on several works across the different domains of Human-Computer Interaction, such as Data Visualization, Swarm User Interfaces, and Data Phsyicalizations. This chapter discusses and organizes those works.

2.1 Data Visualization

In a foundational work on Data Visualization, Cleveland and McGill [10] proposed a ranking of perceptual tasks in data visualizations and found that some aspects of visualizations, like position, provide information with more clarity than others, like volumes. These rankings persist as "channels" to this day and give an understanding of how people are able to perceive quantitative and qualitative information from visualizations [15]. Tufte presented guiding principles for data visualization, providing a clear outline for improving future visualizations. Brehmer et al. [16] defined a multi-level typology of abstract visualization tasks, exploring *why* a task is performed, *how* it is performed and the methods used to conduct the task, and *what* its inputs and outputs are. In her literature review, Sandouka organized the tasks most often used in interactive visualizations, such as identification, comparison, and categorization, as well as interactions that users have with data, such as reconfiguration and selection [17]. A Survey of Dynamic Graph Visualization conducted by Beck et al. found a variety of approaches to visualizing changing relationships between objects [18]. NetworkBots applies the tasks found by Brehmer and Sandouka to an interactive network physicalization.

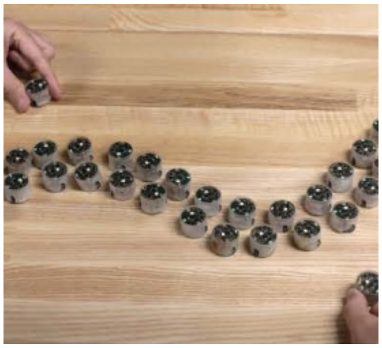
A popular way of visualizing network and graph-based data is to use an Eades Force-Directed graph, which simulates repulsion and attraction between nodes to spread them out over a 2D space [19]. Spritzer et al. built on Force-Directed graphs to allow users to directly interact with network visualizations through operations such as unions and intersections [20]. NetworkBots simulates a Force-Directed graph, and directly enables users to interact with it in order to accomplish different tasks by physically embodying it with actuated hardware.

2.2 Tangible and Swarm User Interfaces

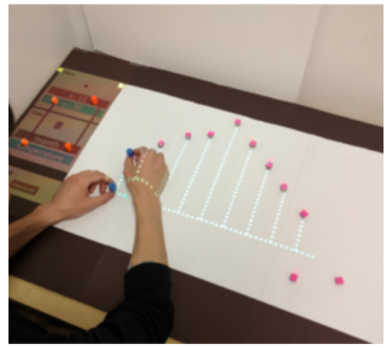
The foundational papers of Tangible Bits [21] and Radical Atoms [22] proposed the idea of Tangible User Interfaces. While traditional Graphical User Interfaces (GUIs) display information through graphics on a screen, Tangible User Interfaces (TUIs) embody digital information in the physical world, employing objects, surfaces, and spaces. TUIs allow users to physically interact with an interface rather than view it on a screen.

Researchers have pushed the idea of TUIs even further by exploring the possibilities of using independent moving units as a collective in a developing class of interfaces known as Swarm User Interfaces (SwarmUIs). The concept of SwarmUIs was first introduced in the Zooids project (Figure 2.1a), which combined wheeled mobile robots along with a light projector to create a tangible user interface where users could interact with individual units to interact with the swarm as a whole [13]. The larger implication of this paper was using individual robots to facilitate interactions with 'stuff' (shapes and materials that can be manipulated) rather than more traditional interactions with 'things' (larger, solid objects).

Reactile built on this new manipulation aspect of interaction to create a SwarmUI that enabled manipulation and interaction with spatial information (Figure 2.1b) [23]. Users could interact with individual nodes to manipulate shapes, and the nodes could be moved by hidden magnetic coils. Alonso-Mora et al. investigated intuitive ways of controlling SwarmUIs by creating gestures to select subsets of robots and assign target goals and trajectories or to manipulate pre-existing shapes [24]. Kim et al. explored user-defined gesture control interactions for tabletop swarm robots by allowing users to define gestures to control robots in the ways that felt most intuitive to them [25]. They further expanded on their work



(a) Zooids explored the usage of SwarmUIs and interactions with ‘stuff’ [13]



(b) Reactile enabled manipulation and interaction with spatial information [23]

Figure 2.1: Previous works in Swarm User Interfaces

by looking into how SwarmUIs could facilitate fidgeting interaction [26], researching how users would fidget with actuated swarm robots. These papers guided the forms of SwarmUI interaction used in NetworkBots. In particular, Reactile served as a foundation for how NetworkBots allowed users to spatial manipulate swarm robots in order to interact with network-based information.

2.3 Data Physicalization

While Data Physicalizations have existed for thousands of years, their use in the field of Human-Computer Interactions was recognized by Yvonne Jansen in her PhD thesis [27] and expanded upon by Jansen et al. [11] to discuss their growing use to represent data in physical space. In particular, they discuss how data physicalizations act as a tangible counterpart to data visualizations. Data physicalizations have taken an expandingly wide variety of both medium and data sources, allowing for a myriad of physical representations of data. In particular, tangible and shape-changing interfaces like inForm [12] have served as a platform for a new type of interactive data physicalization.

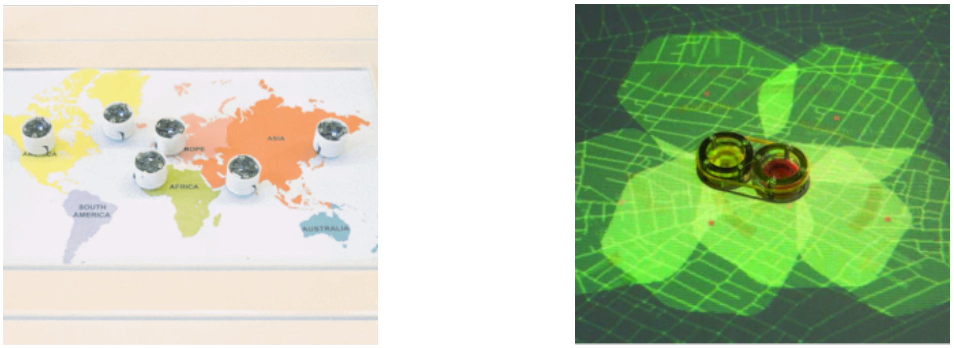
Hornecker et al. developed a design vocabulary for Data Physicalization, defining a language to describe the channels through which data physicalizations can communicate information [28]. Bae et al. have explored how researchers have represented and expressed data through these channels through a wide variety of physical artifacts and interactive mediums [29]. NetworkBots explores the use of visual and haptic channels to communicate information about connections between individual data points, as well as an overall set of data.

Interactive Physicalization

In a particularly foundational work, Le Goc. et al. built on top of Zooids to create representations of data using SwarmUIs, allowing users to interact with individual points of data on tabletops (Figure 2.2a)[14]. Their works explored much of the same design space as NetworkBots, such as using SwarmUIs to physicalize information, focusing on scatterplots and proximity-based encoding, and expressing two-dimensional data points as interactable objects. NetworkBots expands on the discussed future works of Zooids, applying the same concepts to network-based data.

Another key predecessor of my work was PICO, which actuated objects to represent physical data, allowing for the transformation of 'Mechanical Constraints' into 'Computational Constraints' (Figure 2.2b) [30]. PICO used specialized equipment that embedded electromagnets within a table to move objects on its surface. PICO enabled computers and humans to interact in novel ways, enabling interactions that converted abstract and complex tasks into physical processes that users could more easily understand and explore. This work served as groundwork for many of the interactions explored by NetworkBots.

In work similar to my own, Physica used a combination of projection and tabletop mobile robots to allow for a user to tangibly interact with a physics simulation [31]. Building on this previous work, NetworkBots also uses tabletop mobile robots to allow users to interact with networks directly and enable them to apply physical constraints to specific nodes and links. I also employed the same velocity targeting system as seen in Physica in the NetworkBots



(a) Le Goc et al. used SwarmUIs to display 2D information [14]

(b) PICO applied mechanical constraints to create computational constraints [30]

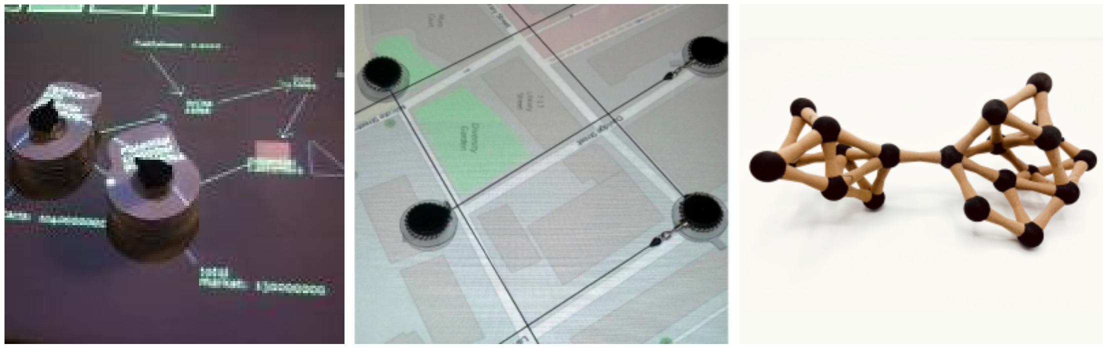
Figure 2.2: Previous works in Interactive Physicalizations

simulation.

Network Physicalization

Network Physicalization has been explored by Tangible Reels [32], Sensetable [33], and Kobayashi et al. [34], but many of them have relied on static objects to physicalize data. Tangible Reels allowed users to create networks of tangible reels to physicalize digital maps (Figure 2.3b). Sensetable enables users to interact with objects on a tabletop surface in order to manipulate a network directly, as well as modify its individual nodes and links (Figure 2.3a). Kobayashi et al. expanded on Sensetable to allow users to place and move pucks to change an IP network simulation directly. Much of NetworkBots was inspired by how these works enabled the direct manipulation of network topologies, as well as parameters of nodes and edges. However, while these works did provide visual feedback through a real-time projected simulation, the static pucks prevented the same haptic feedback that NetworkBots is capable of.

Bae et al. constructed a pipeline to 3D print physical artifacts to represent individual graphs (Figure 2.3c) [35]. Similar to Bae et al., NetworkBots creates a computational pipeline to take a more generalized approach to create an interactive pipeline that can design and



(a) SenseTable allowed people to directly interact with data [33]

(b) Tangible Reels allowed users to physicalize networks using tangible reels [32]

(c) Bae et al. 3D printed physical artifacts to represent networks [35]

Figure 2.3: Previous works in Network Physicalizations

build on physicalizations like Tangible Reels, Sensetable, and Kobayashi et al. [32, 33, 34].

2.4 Summary

NetworkBots builds on a wide variety of works in the fields of Data Visualization, SwarmUIs, and Data Physicalizations. While previous works have explored this intersection, NetworkBots focuses specifically on expanding these works to physicalize networks. From earlier works in data visualization and physicalization, NetworkBots uses visual and haptic channels to explore how abstract data can be expressed through visuals, movement, and force. NetworkBots also adapts many interactions presented by previous SwarmUIs that present interface as ‘stuff’ rather than ‘things’.

NetworkBots explores a gap in the current field of data physicalization and SwarmUIs to physicalize networks through SwarmUIs. Building on interactions demonstrated in previous works in SwarmUIs and techniques of representing data from previous data visualizations and physicalizations, NetworkBots explores how SwarmUIs can transform abstract network-based information into interactive data physicalizations.

CHAPTER 3

LAPTOP-TOIO

Produced and sold in Japan by Sony Interactive Entertainment, toios are a tool used for tangible education and gaming, but they have also served as a valuable prototyping tool in Human-Computer Interaction Research. Despite their use in research and their commercial availability, there exists no consistently used system for controlling toios. The capabilities of toios, including control over their motors, speakers, and LED, are defined in the publically available toio specification [36]. It provides a detailed description of how to send and read messages to toios. However, existing methods require external hardware like Raspberry Pis or are no longer maintained. Previous works like the rust toio crate [37] and the toio-osc repository [38] allow for communication with toios without any external hardware, but neither of them still maintained or support the full toio specification.

Building on the basic framework provided by projects like toio-osc, I developed Laptop-Toio as a toio control platform over two years and multiple research projects to create an open-source method of controlling toios without the use of external hardware. Furthermore, Laptop-Toio was used at AxLab as a platform for easy prototyping and design for tangible human-computer interaction as well as data physicalization. This chapter explores its implementation, development, and the components of its design that furthered its use in Data Physicalization.

3.1 Technical Implementation

3.1.1 Overall Structure

As seen in Figure 3.1, the Laptop-Toio platform consists of two core codebases:

- A Rust server that enables communications between a device (primarily laptops and computers) and several toios.

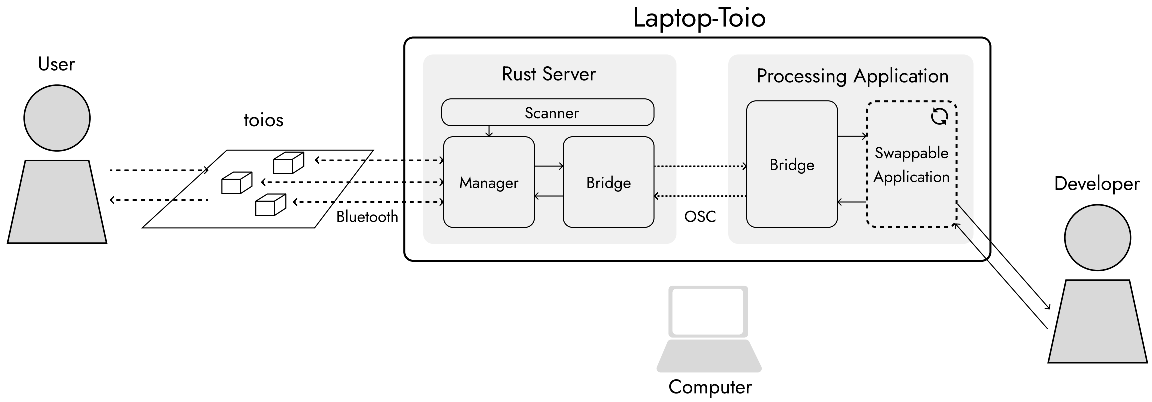


Figure 3.1: The System Overview of Laptop-Toio

- A codebase made in the Processing language that serves as a template to allow for designing and building applications for toios.

While the Rust server is the backbone of Laptop-Toio, the Processing codebase is where the most significant portion of development and designing will occur. The two applications then communicate using the Open Sound Control (OSC) protocol as a bridge.

3.1.2 Rust Server

Laptop-Toio built upon toio-osc, a repository that provided a structure for communicating with a high number of toios with low latency but only implemented two of the commands in the toio specification. Laptop-Toio extends this functionality to the entirety of the specification and provides more direct control over the connection process.

In total, there are three core components to the Rust server: The scanner, the manager, and the bridge. The *Scanner* continuously searches to find new toios to connect to. As the scanner finds new toios, they are passed to the *Manager*, which connects to toios, records the peripheral to send messages, and then spawns a new thread for each toio to listen for new messages. The manager works in conjunction with the *Bridge*, which sends and listens for OSC messages from the Processing application. The scanner, manager, and bridge each sit on their own thread and use channels to send messages back and forth.

3.1.3 Processing Applications

Processing is an application widely used by artists, designers, and coders in interaction design, creative coding, as well as interactive graphics [39]. With a relatively simple structure, users can construct interactive applications. The Processing code template of Laptop-Toio builds on the simple structure of Laptop-Toio to make it as easy as possible to prototype and design interactive interactions between toio robots and application users.

Similar to the Rust server, there is an *Bridge* within the Processing template to send and receive OSC messages. The bridge allows for accessing live information from each of the toios, as well as the capability to send commands. The bridge is connected to the developer-controlled *Application*. Because the Rust Server and Processing Bridge handle the heavy technical lifting, the application template provides flexibility for developers to implement new applications with specific interactions in mind.

3.2 Development

Building on previous projects like toio-osc, Laptop-Toio was developed slowly and evolved over multiple stages through research and classroom use. This subsection highlights the major projects developed with Laptop-Toio, how they pushed its development, and how these developments can be used in data physicalizations. Projects developed with Laptop-Toio are discussed further in Appendices A and B.

3.2.1 Laptop-Toio v1.0

The initial version of Laptop-Toio built upon the structure of toio-osc, expanding its functionality to support the entirety of the toio specification. With the newly added support,

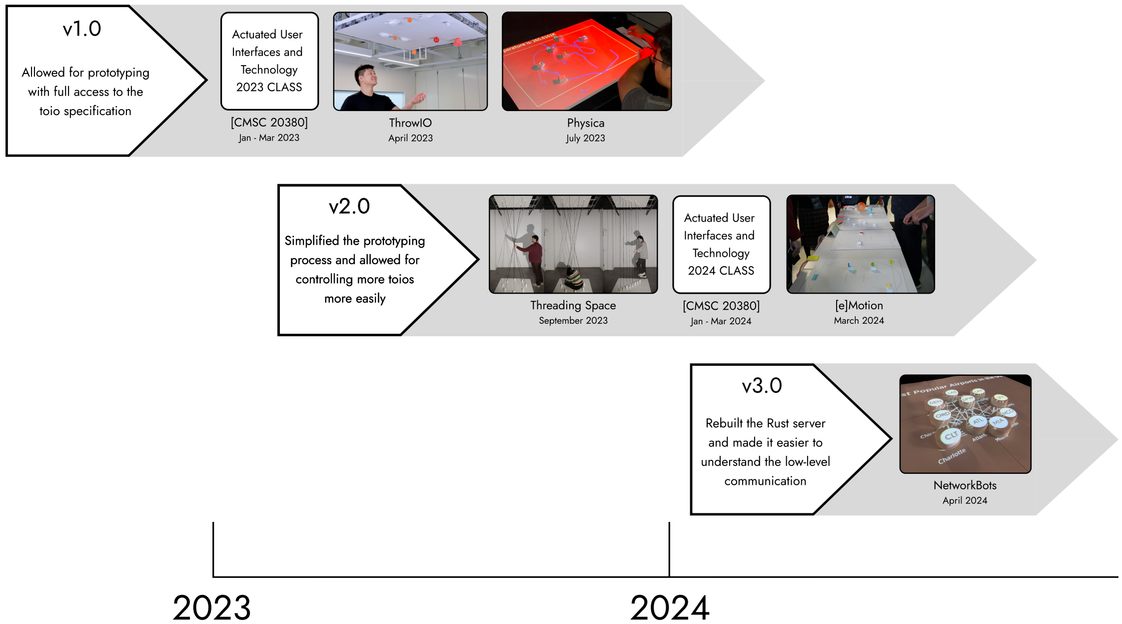


Figure 3.2: The Development of Laptop-Toio and its corresponding projects

Laptop-Toio provided complete control of up to 12 toios, as well as their motors, LEDs, and speakers. It also enabled connecting to individual toios in specified orders, assisting applications in using toios for unique roles, and allowing for multiple toio applications to run in the space without interfering with each other.

This version of Laptop-Toio was used in research projects such as:

- Physica: Physica combined Laptop-Toio with a physics simulation to create a tangible physics simulation that could educate on concepts such as gravitational force, molecular movement, and spring force [31].

- ThrowIO: ThrowIO used toios on overhanging surfaces to create an actuated tangible user interface that facilitated throwing and catching spatial interaction [40].

- Laptop-Toio v1.0 was also used in the 2023 offering of the University of Chicago class 'Actuated User Interfaces and Technology,' where students prototyped their own data physicalizations.

In particular, the initial development of Laptop-Toio explored how it could be used in Data Physicalization. Physica defined a velocity targeting system that provided more direct control over the movement of robots to simulate specific motions. The remaining data physicalizations within the Actuated User Interfaces and Technology class further explored the capabilities of robots in motion.

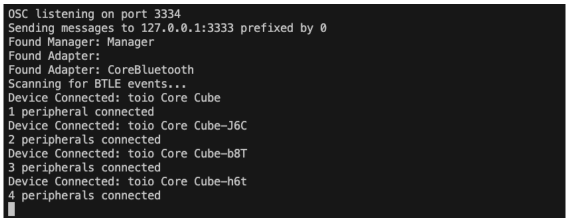
3.2.2 Laptop-Toio v2.0

Laptop-Toio v2.0 was a rewrite of the Processing Template to facilitate the creation of new applications more effectively. It created the bridge abstraction in the Processing template to separate the code that handled the OSC messages, simplifying the prototyping process for developers. This made it easier to control multiple toios without needing to consider the connection to each individual toio.

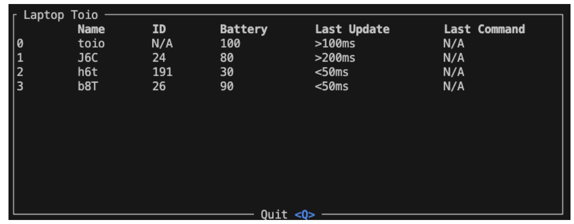
This version of Laptop-Toio was used in Research Projects such as:

- Threading Space: Threading Space placed toios on the floor and the ceiling to construct a kinetic string sculpture that outlined planes and volumes in space, creating pseudo boundaries and playing with the viewer's natural delimitation.
- RE-Motion: RE-motion allowed users to record and playback toio motions. By combining recorded robotic motions with constructed objects, users could express emotions through moving robots [41].
- Laptop-Toio v2.0 was also used in the 2024 offering of the University of Chicago class 'Actuated User Interfaces and Technology.'

By abstracting away OSC communication and simplifying the process of collecting and sending information to toios, this version of Laptop-Toio made it easier to design actions that used contextual information. This also simplified the velocity targeting system and allowed



(a) The old notification stream for the Laptop-Toio Rust Server



(b) The new terminal user interface for the Laptop-Toio Rust Server in v3.0

Figure 3.3: Comparing the Terminal Line Interfaces of Laptop-Toio

for the collective smooth movement of multiple robots, as seen in both Threading Space and RE-motion.

3.2.3 Laptop-Toio v3.0

The latest version of Laptop-Toio was v3.0, which completely rewrote the Rust server code to allow for monitoring the server more clearly. On the surface level, it presented a new terminal-line interface for monitoring toio connections (Figure 3.3). However, it also replaced a system that hard-coded each individual command and updated with a generalized system to make it easier to upgrade as the toio specification expands. This version of Laptop-Toio was used in NetworkBots. This final version of Laptop-Toio is a powerful tool for turning toios into a platform for designing tangible interactions and data physicalizations.

CHAPTER 4

NETWORKBOTS

This chapter introduces NetworkBots, exploring both its interaction space as well as a detailed description of its implementation.

4.1 Interaction Space

The interaction space of NetworkBots builds heavily on previous works discussed in Chapter 2. In particular, it extends the foundations of interactive data physicalizations and visualizations and the interactions of previous SwarmUIs like Zooids [13] and Reactile [23], and applies them to network physicalization, allowing users to interact with networks physically (Figure 4.1). As seen in Figure 4.2, the Interaction Space is built on *Perception Modalities* and *Network Exploration*, which build upon the channels and tasks discussed in Chapter 2, as well as *Network Interaction*, which explores the tasks that are facilitated through NetworkBots.

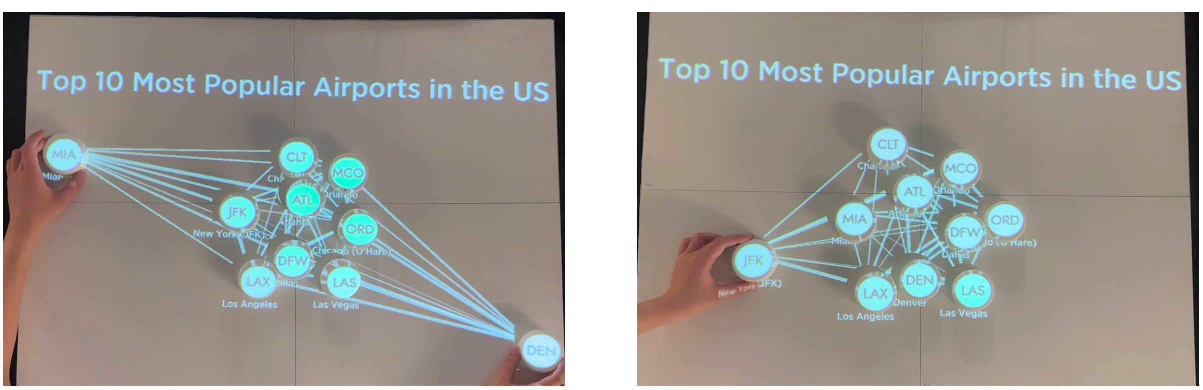


Figure 4.1: Exploring the interactions of NetworkBots

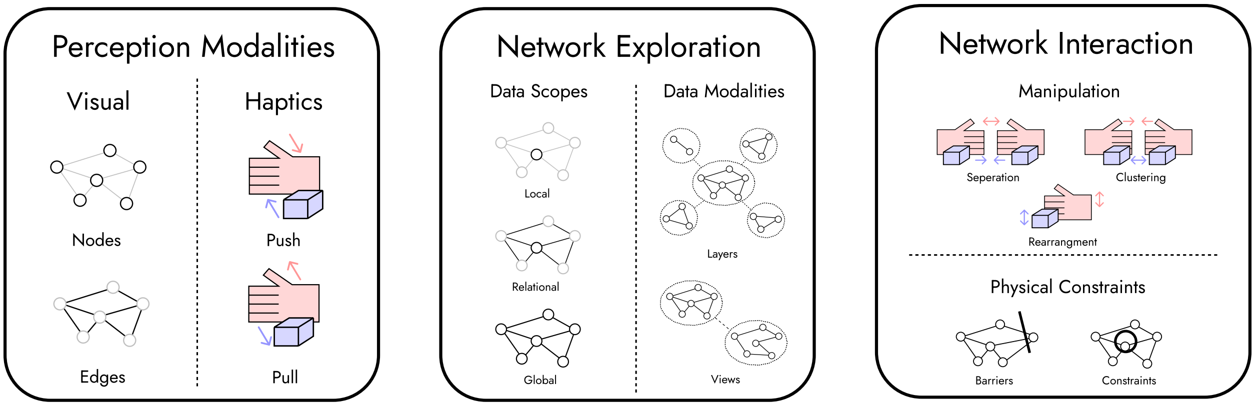


Figure 4.2: The Interaction Space of NetworkBots

4.1.1 Perception Modalities

The perception modalities through which NetworkBots expressed information are through **Haptic Feedback**, such as when a user feels the force applies to a node, and **Visual Feedback** when a user can see how nodes move within the simulation through projected imagery.

Haptic Feedback will mostly occur through the sensations of *Pushing* and *Pulling*. As users separate nodes within a network, there will be a sensation of the nodes pulling on each other. As users push nodes together, they will feel the sensation of the nodes pushing each other apart. These haptic sensations communicate the connection between the nodes physically. Visually, NetworkBots will provide information about the *Nodes* and *Edges*. NetworkBots can project images and text onto nodes, allowing users to understand the node in more detail through the associated image, as well as details like names and properties. Edges are also projected onto the simulation, with varying thicknesses for different weights, allowing users to understand visually which nodes are connected and how heavily. NetworkBots is uniquely capable of providing haptic feedback, applying the foundation created by works like Zooids [13] and Reactile [23] to transform network visualizations into network physicalizations.

4.1.2 Network Exploration

At its core, the NetworkBots system enables an understanding of a network at the different **Scopes** of understanding Data. At the *Local* level, users can understand details about specific nodes, such as their names and any visuals. At the *Relational* level, users can understand the relationships between nodes and their immediate neighbors, such as if two nodes are connected and by how much. On the *Global* level, they can understand the overall topography of the Network, such as its overall density and shape, which can reveal aspects such as clusters or separated nodes.

NetworkBots also provides exploratory structures in the form of **Data Modalities**. These modalities offer the ability to convert more extensive networks into structures that can then be explored in more detail by users. NetworkBots can partition a network into multiple *Layers* that can be individually explored in more detail. The top layer in this structure presents an overall view of the data points, which can each be explored separately in more detail. Alternatively, different perspectives of the same network can be presented as distinct *Views*, such as exploring different potential connections between the same nodes. This enables users to physically experience the differences as the graph is expressed in other views.

4.1.3 Network Interaction

While traditional digital network visualizations allow users to interact with individual nodes through mouse and keyboard or touchscreens, NetworkBots is uniquely capable of enabling different network comprehension tasks through direct interaction with individual or multiple nodes. This is conducted through **Manipulation** and **Physical Constraints**. Manipulation is when users directly modify the graph and feel force feedback in return. *Clustering* occurs when users push together multiple nodes into a collective group to understand the cluster in the context of the larger graph. *Seperation* is when users pull on a single node

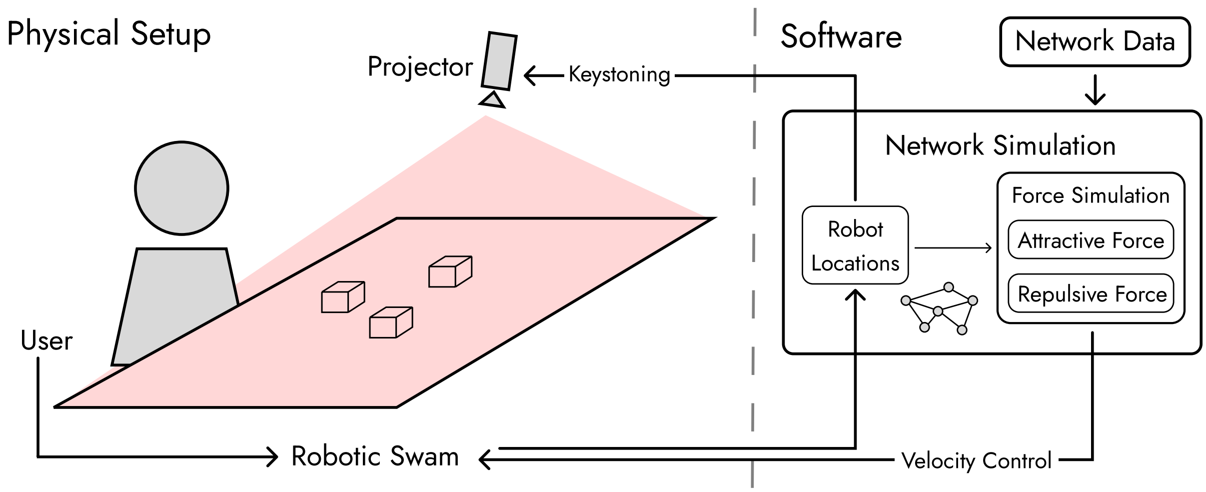


Figure 4.3: The Technical Implementation of NetworkBots

to separate it from the larger network to understand the nodes in context. As users push on nodes to cluster them, they'll feel the nodes pushing back, and as they push on nodes to separate them, they'll feel the nodes pulling back. *Rearrangement* provides the ability for a user to transform the shape of the overall graph by physically manipulating and moving multiple nodes to transform the topography of the network. Because NetworkBots allows users to interact with multiple nodes simultaneously, users may conduct multiple network tasks at once or use them to perform other tasks, like sorting or searching.

Physical Constraints can take the form of *Barriers* or *Constraints*. Barriers completely partition an individual node or a group of nodes from the larger overall graph. Constraints limit the range of motion of individual nodes. The NetworkBots simulation converts these physical constraints into 'computational constraints'.

Figure 4.4: The Physical Setup of NetworkBots at the Museum of Science and Industry

4.2 Technical Implementation

4.2.1 Physical Setup

As seen in Figure 4.4, The physical setup of NetworkBots requires 10-12 toios, a toio mat, and a projector. The toio mat allows the toios to localize and transmit their current location [42]. The projector then overlays a visualization of the simulation onto the toio mat.

4.2.2 Network Simulation

At its core, NetworkBots is a pipeline that physicalizes network and graph-based data interactively. To accomplish this, the application can load a ".csv" file formatted as a 2D array of weights and create a set of nodes with edges with matching weights. For a ".csv" file with n rows and n columns, NetworkBots will then parse that to find the names of all n nodes as well as the weights of the connections to all other nodes. Once the network is initialized, it is passed into the simulation.

The NetworkBots simulation builds on the Eades Force-Directed graph to display the weight of the edges between individual nodes. In order to accomplish this, there exists a

Repulsive force that repels all nodes away from each other and an *Attractive force* that pulls all the connected nodes together towards an ideal distance.

The repulsive force of a node a on a node b is:

$$\frac{c_{rep}}{||a-b||^2} \cdot \overrightarrow{ab}$$

The attractive force of a node a on a node b with an ideal distance $l_{a,b}$ is:

$$c_{attr} \log \frac{||a-b||}{l_{a,b}} \cdot \overrightarrow{ab}$$

where c_{rep} and c_{attr} are the repulsive and attractive constants used to scale the output physicalization. The advantage of this system of equations is that it only requires the current positions of each of the nodes at a given moment to calculate the next position, allowing the simulation to continue working even as users intervene.

4.2.3 Physicalizing and Visualizing the Network

The NetworkBots simulation is presented to the user *physically* and *visually*. Physically, the toio robots move around a surface, following the position of the nodes within the simulation. Visually, the simulation is then projected onto the nodes so that users can understand what each node represents.

To physicalize the simulation, the location data of the NetworkBots simulation is continuously sent to each toio. The toios then use the same velocity targeting system employed in Physica [31] to constantly move to their moving target. To synchronize the physical location of the toio to the simulated location of the node, the simulated force controls the motor speed of the toio robot, while the toio robot location is used in the simulation.

A visualization is then overlayed over the robots using the Keystone library in Processing [43] to handle projection mapping. By calibrating the four points of the projection to the four points of a toio mat, the orientation of the simulation projection will be corrected to match the toio robots. Using the live location of the robots, the visualization can then be projected directly on top, allowing users to associate nodes with their corresponding data points.

4.2.4 Supporting Tangible Interactive Functionality

Because the simulation runs continuously, moving any node to another location will repel local nodes while attracting any connected nodes. This means that the simulation will be automatically updated as users move around the toio robots, creating a seamless experience. Because the force is simulated in real-time, whenever a user applies force to the robots, they will feel the force in return. Similarly, as users place physical constraints, the position of the toios is also communicated to the network simulation, converting the physical constraints into computational ones, as the simulated node will be constrained in the same way as the toio.

Users can also interact with the toio robots by pushing down on them, triggering their onboard button. Using this button, it is possible to register short and long presses, which can be used for displaying more information or even transforming the network. These interactions are application and context-specific and are discussed further in Chapter 5.

CHAPTER 5

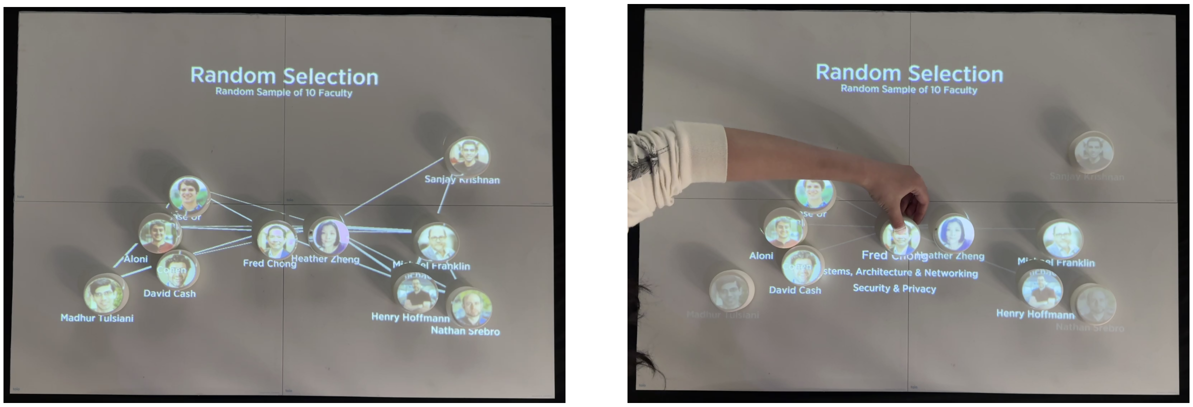
APPLICATIONS

A key aspect of NetworkBots is its generalizability in terms of its representation flexibility and possible interactions, which are grounded in versatile affordances. In its simplest form, NetworkBots allows users to explore the connections between individual nodes within a dataset, but it can also enable more structured exploration. This chapter focuses on different applications of the NetworkBots system and examples of its interaction space.

5.1 Network of Research Faculty

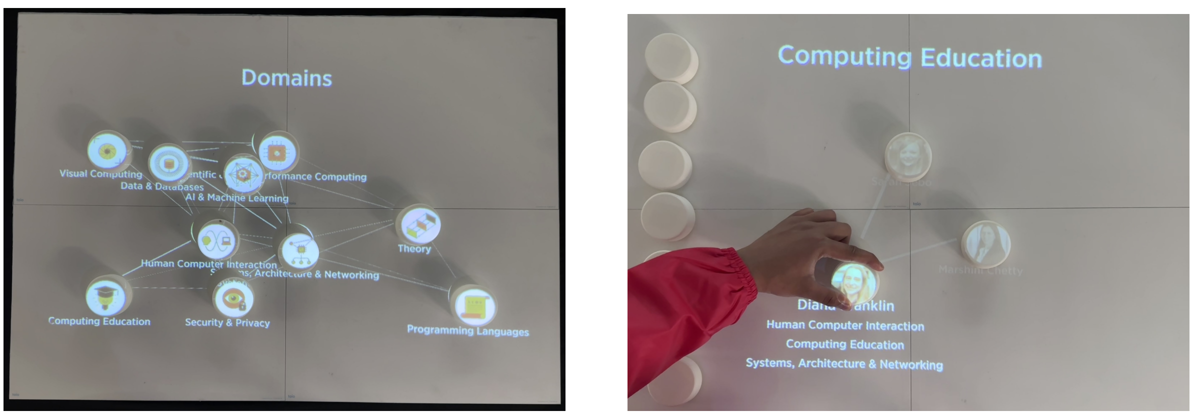
An example of a simple graph physicalized with NetworkBots is shown in Figure 5.1a, which displays a network of ten randomly selected University of Chicago Department of Computer Science Professors. An image of each professor, as well as their name, is directly projected onto each node. The connections between each individual node represent the overlap in the domains of the two professors. *i.e.* If two professors are collectively in four domains but share only two, they share an overlap of 0.5. The higher the overlap between two professors, the stronger the force is pulling them together. In Figure 5.1b, we can see that when a user presses down on an individual node, they can explore its information in more detail, listing all the domains of a selected professor, as well as all directly connected professors.

However, a random sample limits the understanding of the overall network. NetworkBots also provides data modalities to further structure the exploration of the data. In this case, the data can be placed into a larger structure of *layers*, where data points in the top layer can then be explored in more detail, as seen in Figure 5.2. Here, a user first interacts with the domain layer, where they are presented with the ten different domains of the Computer Science Department at the University of Chicago, as seen in Figure 5.2a. They can then press down on an individual node to explore the domain in more detail in the professor layer, where



(a) An Interactive Graph in NetworkBots (b) Exploring a data point in NetworkBots

Figure 5.1: Exploring a simple graph of CS professors at UChicago



(a) Physicalizing the Domain Layer (b) Physicalizing the Professor Layer

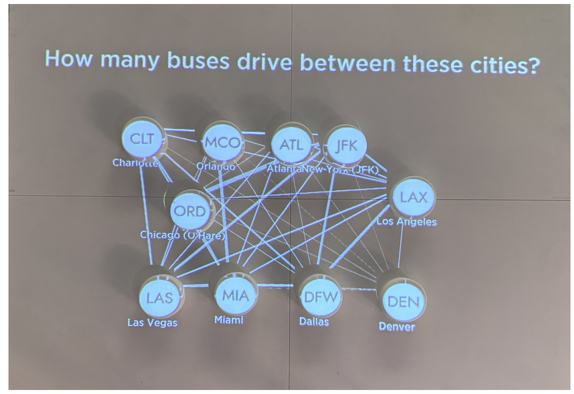
Figure 5.2: Exploring a layered graph of CS professors at UChicago

the individual professors in a domain are visible, as seen in Figure 5.2b. Extra robots on the surface will automatically move to the side to avoid collisions with the remaining nodes. This layer structure provides users a way to interactively explore connections between domains, both on the domain level and to examine specific connections inside a domain.

5.2 Network of Transportation Data

An alternative to providing a layered structure for exploring data in NetworkBots is to create separate *views* of the same data. Each view can offer a different perspective on the same

(a) Interacting with a physicalization of the number of flights between the ten largest U.S. airports

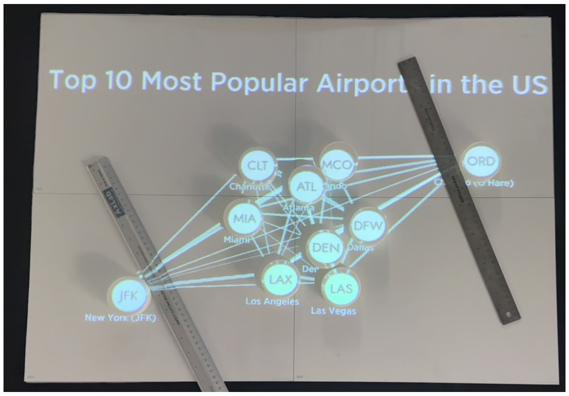


(b) A physicalization of the number of buses traveling between the corresponding cities of the ten largest U.S. airports

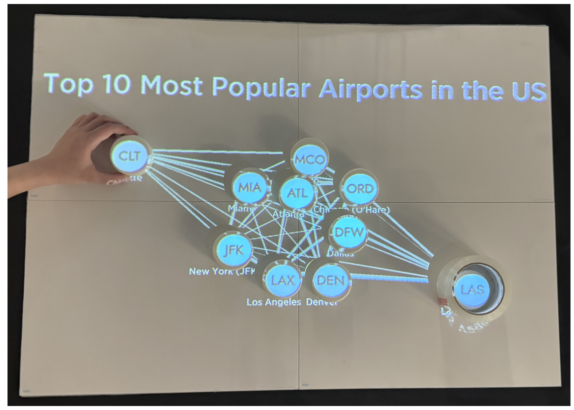
Figure 5.3: Exploring Graphs with Multiple Views

dataset, such as different connections between the data points. Figure 5.3 shows a single dataset displayed with two distinct views: Figure 5.3a physicalizes the number of flights between the ten U.S. largest airports, while 5.3b physicalizes the number of buses traveling between their corresponding cities. In the first view, the force pulling nodes together is stronger if two airports share more direct flights, while in the second, the force is stronger if their corresponding airports share more direct buses. Allowing users to swap between different views enables them to experience and compare the differences within the data physically. In fully connected graphs like this one, NetworkBots also provides the unique ability to understand the tension in the graph, feeling when a graph can't be expressed in an optimal way due to connections to other nodes. As Figure 5.3a also shows, whenever a user presses down on any of the data points, they can see the exact number of flights and buses connecting the nodes in further detail.

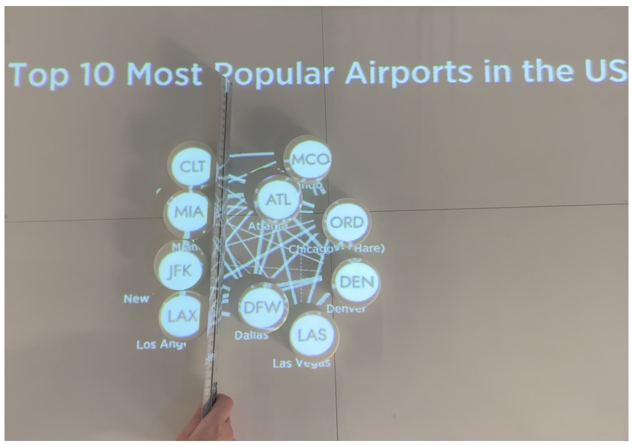
Users can also apply physical constraints to the networks as computational constraints. As seen in Figure 5.4a, *Barriers* allow users to separate data points from the rest of the network. Here, a user places two rulers on the surface to separate the JFK and O'Hare airports from the rest, creating a 'gradient' as the remaining airports position themselves



(a) Constructing physical barriers with NetworkBots



(b) Applying physical constraints with NetworkBots



(c) Partitioning a network with a ruler

Figure 5.4: Using Physical Constraints to Interact with NetworkBots

relative to the force applied from each constrained point. This can act as an easy way to compare how connected airports are to JFK compared to O'Hare. Similarly, users can constrain the range of motion of specific nodes, as seen in Figure 5.4b. This prevents it from moving, even as the rest of the network is physically rearranged. While users can explore a single gradient of connections with a barrier, constraining the motion of one node allows them to create gradients by pulling on the other nodes.

CHAPTER 6

EVALUATION

6.1 Technical Evaluation

For NetworkBots, I found two key points of measurement for technical quality. As the goal of NetworkBots was to communicate to mobile robots to display graph-based information, these key points revolved around factors that would prevent the complete accuracy of the data. These two points of measurement were Communication Speed and Latency. *Communication Speed* is the rate at which the Laptop-Toio system was able to send and receive messages from toios. *Latency* is the lag between the expected position of the toio from the actual position of the toio.

Table 6.1 measures the speed at which Laptop-Toio could process messages from toios. When on a mat, a toio will send a position value at its default max speed of 10ms. However, these messages cannot be parsed at the same speed by the Rust server, leading to some latency. The bottlenecks that could lead to this latency could arise from synchronizing data across multiple threads. To measure this latency, I took the average of five trials conducted at each number of connected toios to measure the speed at which Bluetooth messages from toios were processed. While the lowest measured values of latency occurred with lower numbers of toios, there is no overall general trend in the data.

Network Latency measured the distance between the desired position of nodes within the position and the true position of the toio. This was measured by 20 trials of randomly generated graphs, measuring how well toios could keep up with the moving positions within the graph for 60 seconds. The average distance across the 20 trials between the desired position from the simulation and the true position of the toio in physical space was 0.38cm, which is relatively negligible. Future works could conduct user studies to further quantify the experience and data comprehension of users using NetworkBots.

# Connected	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Average	Std Dev.
1	15	16	16	16	16	15.8	0.44
2	25	28	28	28	30	27.8	1.78
3	18	17	18	18	19	18	0.71
4	18	18	18	19	18	18.2	0.45
5	48	48	45	45	46	46.4	1.52
6	51	40	20	67	26	40.8	18.99
7	25	26	27	27	26	26.2	0.83
8	25	24	23	23	23	23.6	0.89
9	41	35	35	35	35	36.2	2.68
10	41	39	39	39	41	39.8	1.10
11	31	34	34	35	37	34.2	2.17
12	31	32	34	35	35	33.4	1.87
13	32	32	31	30	29	30.8	1.30

Table 6.1: Measured Communication Speed (ms) by number of connected toios

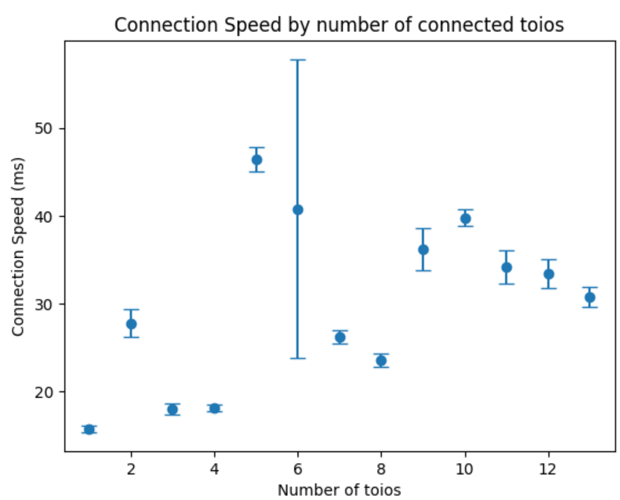


Figure 6.1: The plot of the values and averages found in Table 6.1

(a) Axhibition Demonstration (b) MSI Demonstration

Figure 6.2: Demonstations

6.2 User Reaction from Public Exhibitions

NetworkBots was demonstrated at two locations: The **Axhibition** at John Crerar Library on February 29, 2024, as well as the **Robot Block Party** at the Museum of Science and Industry on April 13, 2024.

6.2.1 Exhibitions

At the Axhibition, users were able to interact with the multi-layer graph physicalization of UChicago CS Professors discussed in Section 5.1. They were also able to use multiple barriers to secure the position of multiple nodes. Passing users explored the domains and professors within them and often explored the connections between professors within the same domain. In particular, users had a tendency to search for the connections of specific professors that they were curious about, exploring the domains to find their target and closely connected professors. They would also use the barriers to constrict the motions of specific nodes and see how they could transform the graph.

At the Robot Block Party, users were able to interact with the physicalization of flights between the ten largest U.S. airports, as discussed in Section 5.2. Groups enjoyed pulling the different nodes apart, with some describing the tension and force between the various nodes as “magnetic.” One user described it as “slime with robots” because of the way it pulled itself back together after different nodes were separated, while another said it felt like the swarm robots had “a will of their own” as the robots moved around in response to interaction. Some even took it as a challenge, as one user attempted to discover the most connected nodes before looking at the actual data. In particular, users expressed that they could feel how connected the graph was.

6.2.2 Limitations

However, users also sometimes found the system unintuitive. Overall, one of the most significant challenges was intuitively communicating the data. In particular, the systems could sometimes provide an information overload, with some saying that it was occasionally challenging to comprehend one-on-one relationships between nodes because each node had an effect on every other node. Networks are inherently abstract, so users didn’t always immediately understand the system’s intention. The continuous movement of the robots also led some to believe that there was a live component to the information, as they would move even without interaction. Others assumed that the system was representing geospatial data in some way and that the location of the data points was correlated with an actual location in some way.

Misinterpretations of the data caused by the technical limitations of using robots were also caused by the graph sometimes expressing itself in non-optimal ways. In the simulation, nodes could pass through each other as they moved to a more stable equilibrium, which would cause collisions when using the robots. In order to synchronize the simulation and the physicalization, the simulation was inherently limited to what was physically possible, which is not a consideration with network visualizations. Another limitation was how lights could interfere with the legibility of the projection, which at times made it difficult to discern the projection's images and texts.

6.2.3 Reflection

At both demonstrations, users, especially young ones, were often mesmerized by the motion. They sometimes initially ignored the data in favor of the kinetic experience of feeling the tension of pulling individual nodes apart and only afterward attempted to actually understand the data. In particular, NetworkBots provided a sense of "exploratory joy" from interacting with the data. It allowed users to have more playful and enjoyable experiences by interacting with the data.

At the exhibitions, there were also times when multiple users would interact with NetworkBots at once, which presents a potential expansion of the interaction space presented in Chapter 4. Users also presented unique datasets for NetworkBots, such as scenarios where NetworkBots could describe biological data. They also introduced unique interactions, such as one user at the Robot Block Party who said that NetworkBots could also present live prices and help plan routes when looking at transportation data.

CHAPTER 7

DISCUSSION

7.1 Future Work

7.1.1 Improved Graph Expression

Currently, NetworkBots best serves as a tool for comparison, as users can directly compare the force they feel between nodes. The system relies heavily on haptic and visual feedback rather than explicitly quantifying the relationships between nodes. However, as seen in Chapter 6.2, this can also lead to misinterpretation of the data. NetworkBots could be refined through user studies to more clearly express information through the existing interaction space. In particular, a more formal user study could test alternatives to Eades Force-Directed Graphs as well as lead to a more robust understanding of the tasks users attempt to accomplish with NetworkBots.

7.1.2 Higher Quantity

A significant limit of NetworkBots was that it was limited to 10-12 toios at once. Future work could expand on this by implementing a system that could connect to and manage a significantly higher number of toios. For example, the complete network of all Computer Science Professors would have 60 nodes, which isn't currently possible to physicalize with NetworkBots. With higher numbers of toios, more extensive networks become possible to physicalize, and new interactions may emerge. In particular, NetworkBots could combine with projects like zorozoro [44], which attempts to control over 200 toios at once but with increased latency and reduced functionality compared to Laptop-Toio. However, by combining both control systems, NetworkBots may be able to maintain a smooth simulation with a much higher quantity of toios.

7.1.3 Advanced Gestures

A current limitation of NetworkBots is that it is limited to only directly interacting with nodes. However, many network tasks may be more abstract and require different gestures. For example, one may want to partition a graph across an axis permanently. While this is partially possible with a physical constraint, as seen in Figure 5.4c, advanced gestures like in Kim et al. [25] could be introduced to NetworkBots to allow users to manipulate in new ways. In particular, future works could explore users moving their hands mid-air to conduct specific network tasks, like freezing specific nodes in place or finding a path between multiple nodes.

7.2 Potential Applications

The potential applications of NetworkBots mostly come from extending its use into specific scenarios. In particular, NetworkBots mostly allowed users to understand static networks. The potential applications presented here mostly extend NetworkBots by discussing non-static networks and tasks other than understanding a specific dataset. It also proposes other networks that NetworkBots may be well suited for.

7.2.1 Dynamic Graphs

Currently, NetworkBots only interactively expresses static networks. However, this same movement could be used to express dynamic graphs, which express network data over time. For example, this could be used to build on projects like Kobayashi et al., [34] which created an interactive static IP Network Physicalization. The applications of expressing dynamic graphs over time could build upon the work done in NetworkBots.

7.2.2 Algorithm Expression and Education

Building on the advanced gestures, NetworkBots could also be used to express graph algorithms. Especially with larger networks, researchers and users could apply algorithms in real time and physicalize their effects. NetworkBots could also be used to educate students on these algorithms with gestures and network modification; students could test out and learn more about graph algorithms. In particular, it could be used to visualize and physicalize exploration, search, and partitioning algorithms, which show up often in both research applications and educational environments.

7.2.3 Applied Datasets

In particular, NetworkBots could be used to express novel datasets that require further exploration. NetworkBots could be expanded into a cross-disciplinary tool to express genomic, biological, or other forms of scientific data. In particular, more context-dependant interactions could be developed so that NetworkBots could serve as a tool for understanding new network information and potentially even lead to new scientific discoveries.

CHAPTER 8

CONCLUSION

This thesis introduced NetworkBots, a system capable of physicalizing network-based information. This new form of physicalization intends to make an abstract form of data more comprehensible and to help people understand an important, albeit sometimes unintuitive, way of expressing data.

To accomplish this, this thesis presented an interaction space of NetworkBots, as well as the implementation of the system. The core aspects of the interaction space were its Perception Modalities, as well as its capabilities in Network Exploration and Network Interaction. It also presented the implementation of its simulation, physicalization, and interaction capabilities. It also discussed the underlying system of Laptop-Toio, a toolkit used for prototyping tangible and actuated human-computer interactions, as well as its development in the course of multiple research projects. This thesis also explored multiple applications and how they demonstrated the interaction space of NetworkBots. In particular, both examples demonstrated the capabilities of NetworkBots in different situations and their context-dependant interactions and network data expression. This thesis then presented a technical evaluation of Laptop-Toio as well as user reactions to NetworkBots to discuss the qualitative and quantitative assessment of the NetworkBots System. This evaluation was then extended in a discussion of the limitations, future work, and potential applications of NetworkBots.

This thesis continues the work done by Euler nearly 300 years ago, exploring how we can more deeply understand the world around us and how we can comprehend it in new and exciting ways. NetworkBots opens up a wide variety of applications for expressing data, everything from the internet to the cells within our bodies. It may even serve as a platform for new discoveries as people build on it to physicalize abstract data in new and exciting ways. Although there was no path to cross all seven bridges in Königsberg, NetworkBots provides a new path of physicalizing networks in a novel way.

REFERENCES

[1] Abdel-Rahman Fowler and Zakaria Hamimi. *Turin Papyrus Map and Historical Background of Geological Work in Egypt*, pages 1–13. Springer International Publishing, Cham, 2021.

[2] Teo Paoletti. Leonard Euler's Solution to the Königsberg Bridge Problem. `https://maa.org/press/periodicals/convergence/leonard-eulers-solution-to-the-konigsberg-bridge-problem`. Accessed 19-04-2024.

[3] Jack Murtagh. How the Seven Bridges of Königsberg Spawned New Math. `https://www.scientificamerican.com/article/how-the-seven-bridges-of-koenigsberg-spawned-new-math/`. Accessed 19-04-2024.

[4] Stephan C. Carlson. Königsberg bridge problem. `https://www.britannica.com/science/Konigsberg-bridge-problem`. [Accessed 19-04-2024].

[5] Königsberg Bridges. `https://mathshistory.st-andrews.ac.uk/Extras/Konigsberg/`. [Accessed 19-04-2024].

[6] Dirk L Hoffmann, Christopher D Standish, Marcos García-Diez, Paul B Pettitt, James A Milton, João Zilhão, Javier J Alcolea-González, Pedro Cantalejo-Duarte, Hipólito Collado, Rodrigo De Balbín, et al. U-th dating of carbonate crusts reveals Neandertal origin of Iberian cave art. *Science*, 359(6378):912–915, 2018.

[7] Frank J. Swetz. Mathematical Treasure: Mesopotamian Accounting Tokens. `https://maa.org/press/periodicals/convergence/mathematical-treasure-mesopotamian-accounting-tokens`. [Accessed 19-04-2024].

[8] Zhi-Wei Liu and Ming-Yueh Tsay. Historical overview of data visualization and its attempts and reflections in lis. 01 2022.

[9] Edward R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, USA, 1986.

[10] William S. Cleveland and Robert McGill. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American Statistical Association*, 79(387):531–554, 1984.

[11] Yvonne Jansen, Pierre Dragicevic, Petra Isenberg, Jason Alexander, Abhijit Karnik, Johan Kildal, Sriram Subramanian, and Kasper Hornbæk. Opportunities and challenges for data physicalization. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 3227–3236, New York, NY, USA, 2015. Association for Computing Machinery.

[12] Sean Follmer, Daniel Leithinger, Alex Olwal, Akimitsu Hogge, and Hiroshi Ishii. inform: dynamic physical affordances and constraints through shape and object actuation. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, page 417–426, New York, NY, USA, 2013. Association for Computing Machinery.

[13] Mathieu Le Goc, Lawrence H. Kim, Ali Parsaei, Jean-Daniel Fekete, Pierre Dragicevic, and Sean Follmer. Zooids: Building blocks for swarm user interfaces. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, page 97–109, New York, NY, USA, 2016. Association for Computing Machinery.

[14] Mathieu Le Goc, Charles Perin, Sean Follmer, Jean-Daniel Fekete, and Pierre Dragicevic. Dynamic composite data physicalization using wheeled micro-robots. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):737–747, jan 2019.

[15] Tamara Munzner. *Visualization Analysis and Design*. CRC Press, 2014.

[16] Matthew Brehmer and Tamara Munzner. A multi-level typology of abstract visualization tasks. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2376–2385, 2013.

[17] Kari Sandouka. Interactive visualizations: A literature review. *MWAIS 2019 proceedings*, 8, 2019.

[18] Fabian Beck, Michael Burch, Stephan Diehl, and Daniel Weiskopf. A taxonomy and survey of dynamic graph visualization. *Computer Graphics Forum*, 36(1):133–159, 2017.

[19] Peter Eades. A heuristic for graph drawing. *Congressus numerantium*, 42(11):149–160, 1984.

[20] Andre Suslik Spritzer and Carla M. D. S. Freitas. A physics-based approach for interactive manipulation of graph visualizations. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '08, page 271–278, New York, NY, USA, 2008. Association for Computing Machinery.

[21] 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, page 234–241, New York, NY, USA, 1997. Association for Computing Machinery.

[22] Hiroshi Ishii, Dávid Lakatos, Leonardo Bonanni, and Jean-Baptiste Labrune. Radical Atoms: Beyond tangible bits, toward transformable materials. *Interactions*, 19(1):38–51, jan 2012.

[23] Ryo Suzuki, Jun Kato, Mark D. Gross, and Tom Yeh. Reactile: Programming swarm user interfaces through direct physical manipulation. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–13, New York, NY, USA, 2018. Association for Computing Machinery.

[24] J. Alonso-Mora, S. Haegeli Lohaus, P. Leemann, R. Siegwart, and P. Beardsley. Gesture based human - multi-robot swarm interaction and its application to an interactive display. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5948–5953, 2015.

[25] Lawrence H. Kim, Daniel S. Drew, Veronika Domova, and Sean Follmer. User-defined swarm robot control. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–13, New York, NY, USA, 2020. Association for Computing Machinery.

[26] Lawrence H Kim, Veronika Domova, Yuqi Yao, and Parsa Rajabi. Swarmfidget: Exploring Programmable Actuated Fidgeting with Swarm Robots. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, New York, NY, USA, 2023. Association for Computing Machinery.

[27] Yvonne Jansen. *Physical and tangible information visualization*. Theses, Université Paris Sud - Paris XI, March 2014.

[28] Eva Hornecker, Trevor Hogan, Uta Hinrichs, and Rosa Van Koningsbruggen. A design vocabulary for data physicalization. *ACM Trans. Comput.-Hum. Interact.*, 31(1), nov 2023.

[29] S. Sandra Bae, Clement Zheng, Mary Etta West, Ellen Yi-Luen Do, Samuel Huron, and Danielle Albers Szafir. Making data tangible: A cross-disciplinary design space for data physicalization. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA, 2022. Association for Computing Machinery.

[30] James Patten and Hiroshi Ishii. Mechanical constraints as computational constraints in tabletop tangible interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, page 809–818, New York, NY, USA, 2007. Association for Computing Machinery.

[31] Jiatong Li, Ryo Suzuki, and Ken Nakagaki. Physica: Interactive tangible physics simulation based on tabletop mobile robots towards explorable physics education. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference*, DIS '23, page 1485–1499, New York, NY, USA, 2023. Association for Computing Machinery.

[32] Julie Ducasse, Marc J-M Macé, Marcos Serrano, and Christophe Jouffrais. Tangible reels: Construction and exploration of tangible maps by visually impaired users. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, page 2186–2197, New York, NY, USA, 2016. Association for Computing Machinery.

[33] James Patten, Hiroshi Ishii, Jim Hines, and Gian Pangaro. Sensetable: a wireless object tracking platform for tangible user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '01, page 253–260, New York, NY, USA, 2001. Association for Computing Machinery.

[34] Kazue Kobayashi, Mitsunori Hirano, Atsunobu Narita, and Hiroshi Ishii. A tangible interface for IP network simulation. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '03, page 800–801, New York, NY, USA, 2003. Association for Computing Machinery.

[35] S. Sandra Bae, Takanori Fujiwara, Anders Ynnerman, Ellen Yi-Luen Do, Michael L. Rivera, and Danielle Albers Szafir. A computational design pipeline to fabricate sensing network physicalizations, 2023.

[36] Sony Interactive Entertainment. toio™Core Cube Specification. `hhttps://toio.github.io/toio-spec/en/`, 2021. Accessed: 2024-01-26.

[37] toio-rs. `https://crates.io/crates/toio`, 2020. Accessed: 2024-01-26.

[38] toio-osc. `https://github.com/MacTuitui/toio-osc`, 2022. Accessed: 2024-01-26.

[39] Processing Foundation. Processing. `https://processing.org`, 2023. Accessed: 2024-01-26.

[40] Ting-Han Lin, Willa Yunqi Yang, and Ken Nakagaki. Throwio: Actuated tuis that facilitate "throwing and catching" spatial interaction with overhanging mobile wheeled robots. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA, 2023. Association for Computing Machinery.

[41] Vali Lalioti, Ken Nakagaki, Ramarko Bhattacharya, and Yasuaki Kakehi. [e]motion: Designing expressive movement in robots and actuated tangible user interfaces. In *Proceedings of the Eighteenth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '24, New York, NY, USA, 2024. Association for Computing Machinery.

[42] Sony Interactive Entertainment. toio™Core Cube Specification. `https://toio.github.io/toio-spec/en/docs/ble_id/`, 2021. Accessed: 2024-01-26.

[43] David Bouchard. Processing and Keystone. `https://fh-potsdam.github.io/doing-projection-mapping/processing-keystone/`. Accessed: 2024-01-26.

[44] zorozoro. https://github.com/Whatever-Co/zorozoro-experiments. Accessed: 2024-01-26.

[45] Ramarko Bhattacharya, Jonathan Lindstrom, Ahmad Taka, Martin Nisser, Stefanie Mueller, and Ken Nakagaki. Fabrobotics: Fusing 3d printing with mobile robots to advance fabrication, robotics, and interaction. In *Proceedings of the Eighteenth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '24, New York, NY, USA, 2024. Association for Computing Machinery.

APPENDIX A
MY PUBLISHED WORKS

This appendix discusses my previous works at the Actuated Experience Lab. During this time, I have published three accepted works with several co-authors. Equal co-authors are marked with asterisks.

A.1 FabRobotics

Presented at TEI 2024, FabRobotics presented a digital fabrication pipeline that combines traditional 3D printing with mobile robots [45]. Integrating these two technologies created new opportunities for 3D printers to fabricate objects quickly and efficiently and for mobile robots to enhance their adaptability and interactivity. To explore this novel research opportunity, we developed a proof-of-concept implementation pipeline, allowing users to execute hybrid turn-taking control of a 3D printer and mobile robots to autonomously 3D print objects on/with mobile robots. The system was implemented with commercially available 3D printers (Prusa MINI) and mobile robots (toio), and we shared various techniques and knowledge specific to fusing 3D printers and mobile robots (e.g., printing mobile robot docks for stable prints on robots). Based on the proof-of-concept system, we demonstrate various application usages and functionalities, showcasing how 3D printing and mobile robots can mutually advance each other for novel fabrication and interaction. My co-authors on this

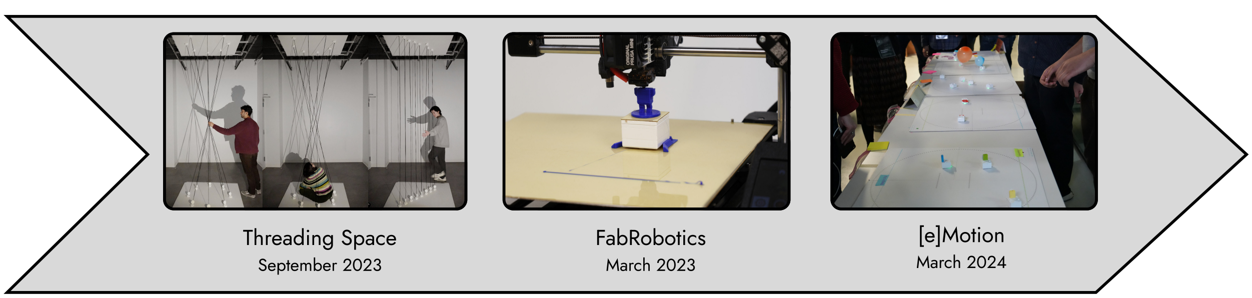


Figure A.1: My Published Works

paper were Jonathan Lindstrom*, Ahmad Taka*, Martin Nisser, Ken Nakagaki, and Stefanie Mueller.

A.2 Threading Space

Threading Space is a kinetic sculpture that explores how spatial perception can be transformed by dynamically and geometrically reconfiguring physical lines of thread. As the threads in motion interact, they become a hypnotic medium for three-dimensional patterns. Through a physical installation and an interactive GUI, Threading Space invites the audience to explore the potential of using swarm robots and line elements to create, morph, and interact with space. This work was presented at Ars Electronica 2023 in Linz, Austria, in September 2023 and at SXSW 2024 in Austin, Texas, in March 2024. My co-creators on this project were You Li*, Emilie Faracci*, Harrison Dong*, Yi Zheng, and Ken Nakagaki

A.3 [e]Motion

Presented at TEI 2024, [e]motion was a workshop focusing on exploring the motive capabilities of robots [41]. As robots inhabit more social spheres, human acceptance significantly impacts their functionality and engagement. The way robotic movement is perceived is crucial to their acceptance in society. However, robotic movement is often a result of function rather than purposefully designed. Working in the continuum between robotic, tangible, and shape-shifting interfaces will enable a deeper exploration of the effects and interpretation of expressive movement. Hence, we proposed [e]Motion, a hands-on opportunity for participants to explore design methods and prototype a variety of expressive movements in robotic and actuated and shape-shifting tangible interfaces. We will collectively reflect on evaluation methods and co-develop a visual vocabulary of motion and emotion, mapping movement more directly to personality and emotion. With this, we aim to foster a practical understanding of expressive movement and how it might affect human acceptance of robots and tangible interfaces. My co-authors in this workshop were Vali Lalioti, Ken Nakagki, and Yasuaki Kakehi.

APPENDIX B
OTHER LAPTOP-TOIO PROJECTS

This appendix discusses other published research works that used Laptop-Toio.

B.1 Physica

Presented at DIS 2023, Physica was a tangible physics simulation system and approach based on tabletop mobile robots [31]. In Physica, each tabletop robot can physically represent distinct simulated objects controlled through an underlying physics simulation, such as gravitational force, molecular movement, and spring force. It aimed to bring the benefits of tangible and haptic interaction into explorable physics learning, which was traditionally only available on screen-based interfaces. The system utilizes off-the-shelf mobile robots (Sony toio) and an open-source physics simulation tool (Teilchen). This paper was written by Jiatong Li, Ryo Suzuki, and Ken Nakagaki.

B.2 ThrowIO

Presented at CHI 2023, ThrowIO was a novel actuated tangible user interface that facilitated throwing and catching spatial interaction powered by mobile wheeled robots on overhanging surfaces [40]. In their approach, users throw and stick objects embedded with magnets to an overhanging ferromagnetic surface where wheeled robots can move and drop them at desired locations, allowing users to catch them. The thrown objects are tracked with an RGBD camera system to perform closed-loop robotic manipulations. By computationally facilitating throwing and catching interaction, our approach can be applied in many applications, including kinesthetic learning, gaming, immersive haptic experience, ceiling storage, and communication. This paper was written by Ting-Han Lin, Willa Yunqi Yang, and Ken Nakagaki.