TANGIBLE EXPERIENCE DESIGN TOOLKIT:

SIMPLIFYING CROSS DEVICE INTERACTION DESIGN

by

Sarthak Marwaha

Bachelor of Technology in Mechanical Engineering, Delhi Technological University, 2019

A Major Research Project

presented to Ryerson University

in partial fulfillment of the

requirements for the degree of

Master of Digital Media

in the program of

Digital Media

Toronto, Ontario, Canada, 2020

Author's Declaration

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF AN MRP

I hereby declare that I am the sole author of this MRP. This is a true copy of the MRP, including any required final revisions.

I authorize Ryerson University to lend this MRP to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this MRP by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my MRP may be made electronically available to the public.

TANGIBLE EXPERIENCE DESIGN TOOLKIT:
SIMPLIFYING CROSS DEVICE INTERACTION DESIGN

Master of Digital Media,2020

Sarthak Marwaha

Digital Media

Ryerson University

Abstract

Cross-device environments are attracting attention from designers working on interactive experiences, as there is an increased desire to create advanced and sophisticated narratives. As these experiences increase in popularity, they use increasingly complex embedded sensor-based environments. These environments deem to be challenging for inexperienced creators as they require knowledge of networked systems. This paper explores the initial design of a toolkit that provides novice designers with an easy to use authoring and development environment for designing cross-device experiences. The toolkit uses Tangible User Interfaces in order to make cross-device development easier to understand for inexperienced designers and provide them with a robust support framework for expanding into developing more complex experiences. The toolkit is tested with a group of potential users. The conclusions from the testing, along with potential future steps, are presented in this paper.

Keywords:

Tangible User Interfaces, Experience Design, Cross-Device Interactions, Interactive Environments, Design Toolkits

Acknowledgements

I would like to extend my sincere gratitude to my supervisor, Dr. Ali Mazalek. Thank you for guiding me through this process and keeping me grounded along the way. I would also like to thank my second reader, Associate Professor Steve Daniels for providing a fresh perspective on the problems I faced and scrutinizing the finer aspects of the project. I would also like to thank Dr. Aneesh Tarun and the members of the Synesthetic Media Lab at Ryerson University for their constant feedback that helped vastly improve the project.

I also need to acknowledge the support of my peers in the MDM program and their constant willingness to help me whenever I faced a challenge. Last but not least, I would like to thank my family and friends for their unwavering support and encouragement throughout my efforts over the last year.

Table of Contents

List of Figures

List of Tables

List of Appendices

Introduction

As new media experiences and digital technologies become more accessible to a broader audience, an increasing number of designers and enthusiasts are looking to utilize these new technologies to create novel interactive experiences. However, unlike programmers, designers have a hard time associating outcomes with required program syntax that requires logic-based construction (Reas & McWilliams, 2011).

Within the personal computing industry, there has been a push towards a world with seamless, ubiquitous computing (Weiser, 1991). Today, most people own and use more than one computing device (Di Geronimo et al., 2016); this opens the avenue to a new class of interactions called cross-device interactions. The premise of such is to be able to use multiple devices to improve personal computing experiences. Further, these interactions can be used to enhance the immersive experience design. Block et al. (2015) suggest that there might be added advantages to using interactive digital objects as a part of experience design as these experiences allow multiple entry and exit points for visitors, unlike traditional exhibits that require reset mechanisms, which make it more accessible regardless of the state of the exhibit. Immersive cross-device experiences are already appearing in the wild. Theme parks are using embedded systems to improve immersion for their visitors; for instance, Universal's Islands of Adventures uses sensors to create working replicas of Harry Potter wands (Acuna, 2014). Similarly, Magic Kingdom's Fantasyland attraction gave guests the ability to interact with objects in the ride queue (Andersson, 2015).

Although young people interact with digital media all of the time, few can create their own games, animations, or simulations. Resnick et al. (2009) phrase this as "It is as if they can read [these digital interfaces] but not write." The same can be said about designers who may understand how cross-device experiences work but do not know how to program them. Cross-device toolkits facilitate authoring of cross-device interactions, but these are usually geared towards experienced developers and toolkits made for novice developers are limited in scope and usually lack the functionality to develop cross-device interactions quickly. Effective design tools should make it easy for novices to get started with design but provide the experts with the ability to create sophisticated designs and suggest a variety of explorations (Resnick et al., 2005).

Tangible User Interfaces (TUIs) provide haptic interaction methods for navigating digital landscapes (Ishii, 2008b). Thus, tangible user interfaces are easier to understand for novice users and improve learning (Weintrop & Wilensky, 2015). This provides an exciting opportunity for a TUI toolkit that fills in the gap as an authoring environment for designers and novice developers to create cross-device interactions. This paper explores the design of a tangible based cross-device design toolkit, which is tested with a group of potential users. The conclusions from the testing and potential future steps are presented in this paper.

Related Works

Background

Although most people are familiar with using digital technologies, only a few people are comfortable creating and designing new media experiences. Even with various accessible resources, most people find programming languages challenging to use and overwhelming (Resnick et al., 2009). Cross-device interaction design and TUI are active research areas. As the toolkit aims to provide a tangible authoring environment for designing cross-device interactions, this section reviews software-based cross-device design toolkits followed by TUI, focusing on the use of TUI for teaching programming languages.

Cross-device interactions

The premise of cross-device interactions is that users can pair and combine multiple devices and interact with them seamlessly. These interactions provide numerous possibilities for designers and developers to improve the way we use our devices today. Current cross-device experiences are simple and usually are limited to an asynchronous approach to utilizing multiple devices. These interactions can be used in a variety of domains and can be designed to fit the required use case; they could involve a set of displays paired together (Schreiner et al., 2015) or the ability to move files between devices by dragging them across space (F. A. Marco et al., 2011) to provide a ubiquitous experience across personal devices.

While useful, these experiences currently do not use the full potential of the multitude of devices that a user might have at their disposal. Houben et al. (2017) discuss opportunities and challenges for developing cross-device interactions in an effort to simplify the multi-device ecologies. Devices and systems today are built for single-user, single-device scenarios; thus, connections between multiple devices together are challenging to develop without certain pre-programmed behaviours and knowledge about all possible configurations. This process can be a significant challenge for designing cross-device interactions as they usually require a full-stack development process. This barrier to entry can be especially tricky for designers and developers who are new to developing networked device systems.

Cross-device interactions in immersive experience design

Over the years, experience designers have increasingly been designing 'interactive spaces' and installations based on spatial interaction (Hornecker & Buur, 2006). These interactive spaces require responsive digital systems embedded in the real space that can offer people opportunities to interact with the space and react to these interactions (Ciolfi, 2004). These 'interactivated spaces' can use sensors to enhance the interactivity of spaces through auditory, visual, haptic or kinetic feedback (Bongers, 2002). Designers that leverage interactive environments can create reactive experiences that are dynamic and enjoyable for the participants (Rubidge & MacDonald, 2004; Ventä-Olkkonen et al., 2014). Much of this work indicates that using spatial interactions and embedded systems to create interactive spaces enhances visitors' experience. Building these experiences requires tools that allow designers to rapidly prototype across the variety of devices that may be used in these interactive spaces (Myers et al., 2000). However, as mentioned earlier, current cross-device development platforms lack easy to use development tools and supporting infrastructures which makes development especially difficult for novice developers.

Development platforms for cross-device interactions

As the need for interactive spaces grows, more designers would need to learn how to program these experiences. There are various examples of cross-device toolkits that attempt to make developing cross-device interactions easier by simplifying the design of interactions between displays, mobile devices (Casteleyn et al., 2014) and wearables (Chi & Li, 2015). Conductor, for instance, is a framework that allows designers to create experiences where users can share information and chain tasks across devices (Hamilton & Wigdor, 2014). XDKinect provides a framework to facilitate the development of cross-device interactions using a Kinect to act as an interface for these interactions (Nebeling et al., 2014). SurfaceConstellations provides a modular hardware platform for linking multiple mobile devices to create a cross-device workspace (Marquardt et al., 2018). However, these toolkits support only a limited set of devices and interactions. Interactive experience designers may need to use an assortment of sensors and mobile devices. Resnick et al. (2005) suggest that effective design tools should make it easy for novices to get started while allowing experts to work on sophisticated projects. They also add a third criterion that the tools should also encourage a wide variety of explorations. They use the terms low threshold, high ceilings and wide walls to define these concepts. The problem is that it is hard to design tools that can meet all three criteria.

Thus, there is a need for a robust universal toolkit that supports a variety of devices. An example of such a toolkit is the Responsive Ecologies toolkit (REtk) that aims to provide a rapid prototyping environment for heterogeneous networked devices (Tarun et al., 2016). REtk builds upon the ROSS API (Wu et al., 2012) to provide an authoring environment that allows developers to design cross-device interactions as hierarchically nested structures. SoD-Toolkit facilitates developing multi-device applications using off the shelf sensors and is supported by a set of widely available libraries (Seyed et al., 2015). While these tools tackle the last two problems of high ceilings and wide walls, they still require the users to have some prerequisite knowledge of networked systems and cross-device interactions.

Tangible interfaces

Tangible user interfaces allow users to interact with physical objects, surfaces and spaces to manipulate digital information (Ullmer & Ishii, 1997). Many tangible user interfaces are focused on the manipulation of digital information using physical objects. These objects provide a tangible representation of the digital information that makes the information directly manipulated while also providing haptic feedback. This allows users to receive passive feedback instead of waiting for digital feedback that would come from a display (Ishii, 2008a). Due to their flexible nature, Tangible User Interfaces can use physical objects of varying fidelities as metaphors for digital interfaces. These can range from the simple physical representation of stored media in the form of a block (Brygg Ullmer et al., 1998) to complex high-fidelity tangible objects used to enhance VR experiences (Harley et al., 2016; Muender et al., 2019).

TUIs and introductory programming

Research suggests using block-based languages in introductory programming education can help students focus on high-level algorithm creation and improve the understanding of text-based programming languages (Matsuzawa et al., 2016). Today, many popular commercial programming/interactive experience design tools use block/node-based scripting interfaces to provide a visual representation of connections and data flow (Bowell, 2020; Resnick et al., 2009; *TROIKATRONIX*, n.d.). The visual representations of connections are easier to understand and allow inexperienced users to build experiences at par with expert users. While these tools are helpful, they could benefit from the use of TUI.

The physicality and passive feedback offered tangible interfaces offer an exciting avenue into programming and designing experiences as they could facilitate the ease of block-based/node-based programming by recreating these interfaces in the physical world. One example by Horn &

Jacob (2007) describes a technique for implementing educational programming using tangibles. They argue that languages that use tangibles as an alternative to text-based and visual programming would be more appealing and practical in a classroom setting. The results showed that this implementation allowed students to easily create flow-of-control chains and understand the actions they had ‘programmed’.

Similarly, Bouchard and Daniels (2015) describe a system of tangible ‘tiles’ that allows fine art students to understand the relationship between proposed networked experiences and the required syntax, software libraries and hardware that is required. The tiles significantly improve the understandability of the code as they gain confidence by developing code samples.

Flowblocks (Zuckerman & Gal-Oz, 2013) is another such toolkit that uses TUI to provide a modelling and simulation environment. The study sets out to test the effectiveness of TUI vs traditional graphical interfaces by creating a GUI version of the flowblocks toolkit. Interviews with their users revealed that the users preferred the TUI based version over the GUI as it enabled physical interaction, provided rich feedback and high levels of realism. These properties made TUIs more enjoyable and stimulating (Table 1). These attributes of TUI make them preferable for users even though TUI fare worse in some objective attributes such as efficiency.

Using the easily understandable interfaces provided by tangibles and the extensive development support offered by cross-device toolkits, it is possible to design a TUI based authoring and development environment for designers looking to design interactive spaces using cross-device interactions. Such a development environment would help designers who may not be familiar with the nuances of cross-device environments.

Attribute	TUI	GUI
Ease of Learning	High	Low
Efficiency	Low	High
Physical Stimulation	High	Low
Realism	High	Low
Enjoyment	Very high	High

Table 1: Attributes of the TUI and GUI versions of FlowBlocks, as reflected in interviews with users (Zuckerman & Gal-Oz, 2013)

Frameworks for Tangible User Interfaces

In order to design a user-friendly TUI, it is crucial to understand the existing design frameworks for TUIs and how they can be used to drive the toolkit's design. Current graphical interfaces (GUI) use the WIMP (Windows, Icons, Menus, Pointers) structure for defining user interactions. However, as we move towards using physical objects to manipulate digital spaces, these metaphors introduce seams and discontinuities into the workspace (Billinghurst, 2003). There are several models that define the interactions and structure of TUIs. The MCRpd (Model-Control-Representations) model uses the MVC (model-view-control) model for GUI and redefines it for TUI (B. Ullmer & Ishii, 2000). The PAC paradigm structures the interactions in the form of presentations, abstractions and controls (Coutaz, 1987). Tokens and Constraints (TAC) systems utilize spatial and relational mapping of physical objects to encode and manipulate digital information on interactive surfaces (Shaer et al., 2004; Brygg Ullmer et al., 2005). These frameworks can be used to extend the design of TUI by improving the capabilities of tangible objects or the sensing environments. For instance, TAC interactions have been further expanded by the use of active tokens that can dynamically modify their connections, spatial or relational, to other objects (Klum et al., 2012; Valdes et al., 2014, 2014; Zigelbaum et al., 2007).

Design Process/Method

As detailed above, using tangibles has a clear benefit towards improving visualization of code syntax for novice users while it may also make it easier for experienced users to develop more complex programs involving interactions between multiple devices. In order to design the toolkit effectively, I felt it was important to further detail the objectives of the toolkit based on the overarching goal of tangible cross-device development & authoring environment that has low barriers, high ceilings and wide walls. Based on this goal the main objectives of the project are to develop a toolkit that:

- Facilitates easier cross-device interactions between networked devices;
- Allows novice users to create custom experiences using a tangible interface;
- Provide users the flexibility to build increasingly complex experiences as they get used to cross-device interactions;
- Supports a variety of devices 'out of the box'.

The design process will focus on university students in immersive experience design programs as the main user base. This allows the toolkit to serve a user base that has a comparable level of expertise but have varying requirements thus the toolkit can support devices that are used by a majority of new designers. This also provides the opportunity to test the toolkit design with its intended users.

Understanding cross-device interaction design

Cross-device interactions are diverse and differ based on various key factors laid out by Brudy et al. (2019). One of these key factors is device dynamics and relationships; thus, defining the types of devices that the toolkit expects to support is important. Devices in cross-device interactions can be classified in one of three categories. Input devices that take various forms of user input and pass it off to application systems for processing. Output devices can render and provide feedback to the user through visuals, haptics or auditory signals. Mixed devices incorporate both input and output interfaces (Scharf et al., 2013).

This breakdown of devices based on their interaction roles was too abstract to design a toolkit. Thus, It was essential to understand the fundamentals of cross-device interactions and break them down into their key components. The breakdown borrows from the ROSS API designs and the breakdown used by REtk as this toolkit supports a wide variety of devices and scales well to add more functionality. The breakdown uses a simple cross-device interaction as the example where a tap on the phone results in a click on a desktop (Figure 1).

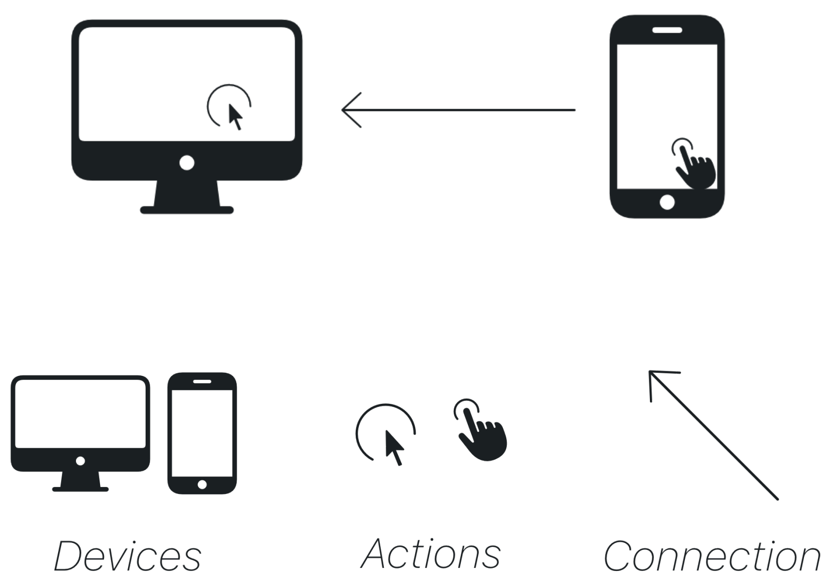

Figure 1: Breakdown of cross device interactions between 2 devices

For this toolkit, Cross-device interactions between two devices are broken into three key components; the devices, the inputs and outputs present on the devices (actions) and a connection protocol that allows the two devices to interact. This forms a hierarchical tree-based structure (Figure 2) where the connection is established between actions, which are children of their respective devices. Complex cross-device experiences can be constructed using multiple simple interactions; thus, the simple interaction was used to design the toolkit.

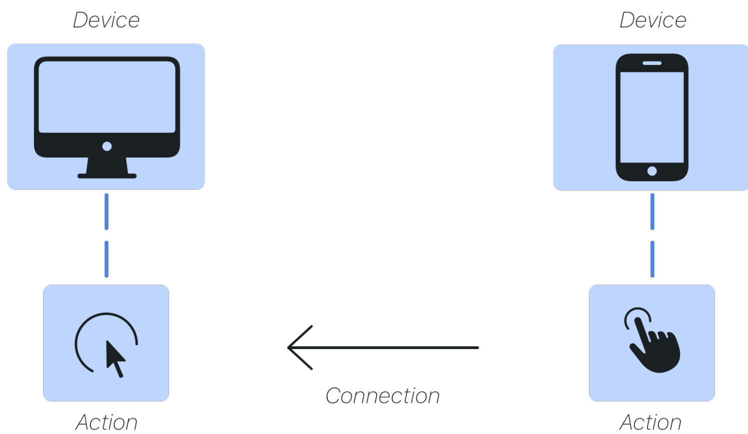

Figure 2: The hierarchical structure for simple cross-device interactions

Formative Research into Tangible Toolkits

Before designing the tangible user interface, I explored existing everyday tangible objects as well as some existing toolkits that utilize tangible user interfaces to understand the affordances various tangible objects offer and the diverse interactions that could these objects could form. These observations were then used to design the tangible objects for the toolkit.

In order to test the affordances offered by everyday objects, I participated in a series of design brainstorm workshops as a part of a course on Embodied Digital Media. These focused on examining three different everyday objects that could provide insight for the purposes of the toolkit - board games, Lego blocks and DIY physical computing components (Breadboards, wires, LEDs, potentiometers, sliders etc.) Every object was judged on the basis of intended use for each object, the expectations a user may have from these objects and how they could be useful in designing the toolkit. The key takeaways from the workshops were as follows:

- The playing areas for the board games provided a limited interaction space that could be useful for designing the toolkit's physical boundaries.
- The Lego bricks provide tactile snap when connecting two bricks together, which could be an analog confirmation of two objects connecting.
- The physical computing tools were highly customizable and modular, which allowed them to be used to design extremely simple or highly complex electronic systems.
- The physical computing tools also offer a glanceable representation of data flow and connected objects using connecting wires.

Additionally, various papers outlining the design of tangible systems we examined as a part of the formative research. These were examined on the basis of the tangibles and the connection interfaces used to find parallels between the breakdown of cross-device interaction detailed in this paper and existing toolkit research (Table 2).

Toolkit	Tangible Objects	Connection Interface
ConnectUs	Cubes with built-in sensor array	Bluetooth
Dr. Wagon	Physical representation of code blocks	Embedded electronics
Tiles that Talk	Tangible tiles	Phone-based application
ToyVision	Designed by Users	GUI based assistant
Paperbox	3D shapes built out of paper using templates	Conductive ink

Table 2: Evaluation of tangible systems as an authoring tool

ConnectUs (Lechelt et al., 2016) uses cubes with built-in sensors and a tablet GUI to provide users with a framework to design IoT systems connected using Bluetooth. This limits the sensors to the ones built into the cubes, but it ensures that the toolkit is easy to use. Dr. Wagon (Chawla et al., 2013) illustrates a very different approach where it uses metaphors from popular programming languages and provides a tangible representation of traditional IDEs that can be used to program a moving robot. The connections are made with electronic components embedded in the blocks. Tiles that talk (Bouchard & Daniels, 2015) uses a set of puzzle piece based tangibles that can be connected to form interactive and networked systems. The connections are made by connecting the tiles in physical space, and then network connections are made by a phone-based application.

While the final two toolkits are not designed for end-user experiences, they provide valuable insight into designing design tools. The ToyVision (J. Marco et al., 2013) toolkit is built upon the Reactivision (Kaltenbrunner & Bencina, 2007) framework and can be used to build a toolkit that facilitates the design and development of tabletop games. The tangible objects are classified into categories based on functionality, and connections are made using a GUI assistant that guides users through the design process. Paperbox (Wiethoff et al., 2013) is a designer focused toolkit that uses low fidelity tangible objects built from provided templates. These templates can be easily assembled into shapes that are typically used in tangible user interfaces. This allows designers to test the graspable interactions between tangible objects quickly. The prototypes can be connected using conductive ink, which provides functionality that may be required in early informal design sessions.

While all the toolkits chose different approaches in terms of abstractions in their tangible objects, they all used some supporting GUI based application to complete the connection process. Toolkits that relied on analog components instead of digital ones allowed users to explore more as they offered the opportunity to experiment.

Design Requirements

The goal when starting this project was to develop a tangible toolkit with low barriers and high ceilings and wide walls. Based on the objectives and the information gathered during the formative research. Some design requirements were drafted. These narrow down the objectives mentioned at the start of this section to fit the toolkit's design process and set some preliminary evaluation criteria.

R1: The toolkit is designed for users who are new to developing cross-device interactions; thus, it should be easy to understand and use for developing simple interactions. This helps the user get acquainted with various components of the toolkit while learning about its capabilities and limitations.

R2: The toolkit should provide a glanceable visual representation of the connections between the devices and their effect on these devices. In order to fulfill this requirement, It is necessary to be able to identify the actions that belong to a device.

R3: The toolkit should support a robust set of interactions between some fundamental devices with the ability to further add to these interactions easily. In the pursuit of high ceilings and wide walls, the toolkit should also be able to support a vast number of inputs/outputs connected to a single device, with each of these having their own interactions along with a framework to modify these interactions to the user's needs.

Fulfilling these requirements should provide a strong foundation for novice users to develop cross-device interactions and provide them with a path to designing and implementing more complex systems with a variety of devices.

Prototype Design

As the primary users for the toolkit will be designing interactive experiences, the initial device support was driven by the frequently used devices in such environments. A large portion of interactive experiences use some combination of an embedded sensor-based environment (Arduino, RaspberryPi and other similar systems), portable user devices and fixed displays. Thus, Smartphones, Desktops and Arduinos were chosen as the initial devices for the prototype. Smartphones and Arduinos offer diversity in terms of sensors and user interactions, while desktop environments can serve as powerhouses to control resource-heavy devices such as interactive displays. These devices were broken down into their respective inputs and outputs, and meaningful connections between the devices were identified (Appendix A). In order to provide a logical visual representation of the device relationship and data flow, a modified version of the cross-device structure laid out for ROSS API (Wu et al., 2012) was chosen. Wu et al. propose a nested hierarchical structure representing all devices present in a given space where connections are formed between devices.

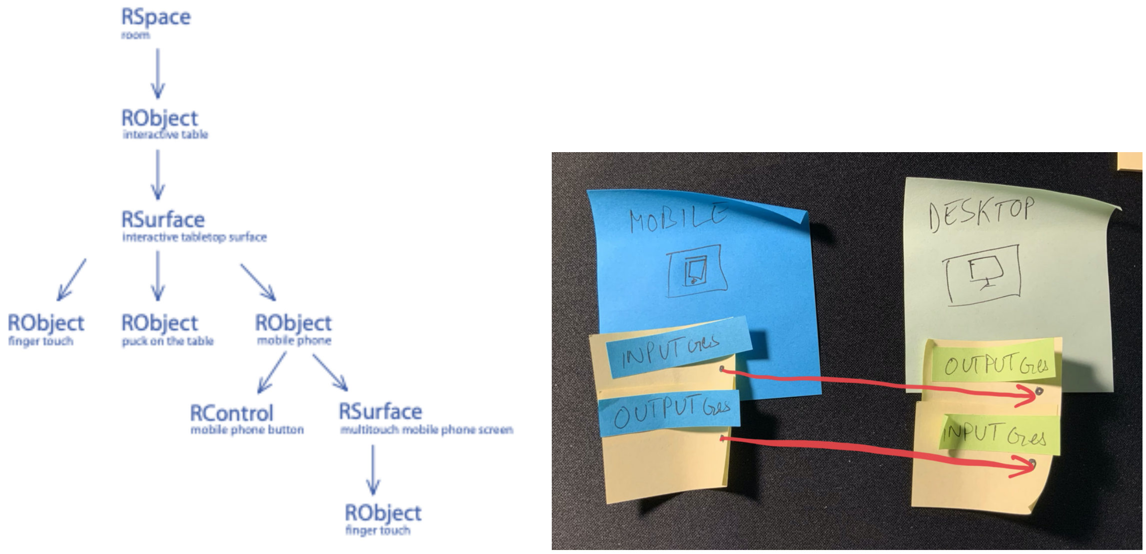

Figure 3: Nested Hierarchical structure for multi-device environments laid out for ROSS API (Wu et al., 2012) compared to the one described by the prototype

This nested hierarchical structure appears to fit well into a tangible toolkit as it provides a representation of devices as well as their connections at a glance. However, the connection interface proposed in ROSS API did not translate well to tangible space (Figure 3). Connections represented between devices did not illustrate the actions that were being used for a particular interaction. These connections can be represented with a lower level of abstraction, where the connections between specific actions are visible to the user in the form of interactions that the user would perform. This adds another design requirement:

R4: There should be a clear distinction between (1) the interactions and the effect it has on other devices and (2) the connection between a device and its respective inputs/outputs and interactions.

Tangible Experience Design toolkit

Using the observations from early prototypes and existing toolkits, the Tangible Experience Design toolkit's initial design is outlined below. There are four key tangible components of the toolkit - Device blocks, Interaction Containers, Interaction cards, Connections that can be used to build a visual framework for the interactions between the devices. The initial version of the toolkit supports three devices; Desktop, Mobile, Arduino, with two interactions per device (Appendix B). The tangible interface is supported by a graphical interface that allows the user to set the properties of the interaction cards and modify them to their preference.

Device Blocks

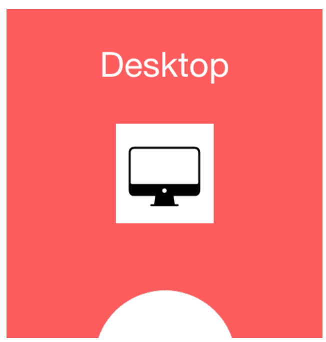

Figure 4: Device block for Desktops

The device blocks (Figure 4) act as the major nodes for authoring the interactions. They feature a puzzle-piece derived interface along the bottom edge. Gestures or interactions can be modeled by connecting the required tangible objects to these device blocks.

Action Containers

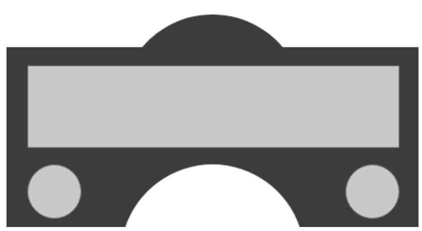

Figure 5: Interaction container

In an effort to allow rapid prototyping of interactions using different sensors while using a limited number of tangible objects, the inputs and outputs for the interactions are housed in generic interaction containers (Figure 5). The interaction containers can be connected to the device blocks using the puzzle-based interface. Actions are defined using smaller cards that can be slotted into the interaction containers. The interaction containers also act as a connection interface between the devices. Each interaction container has two connection points that let the user connect one interaction card to another forming a connection between the two devices for the given set of interactions.

Action Cards

Figure 6: Action card for a mobile device denoting the swipe action

The action card follows the same design language as the device blocks (Figure 6). They have background colours which can be used to reflect the compatibility with various devices. The interaction cards are also designed to be modified and adjusted to the user's needs as they get more comfortable with developing cross-device interactions. As the users get more experienced with cross-device interactions they could also develop their own interactions and easily extend the ones supported by the base toolkit.

Connections

The toolkit required two easily identifiable metaphors to represent the types of connections between the objects. First, the intra-device connections between interactions and their parent devices and second, the connections between two interaction cards connected to different devices that would define the interactions (Figure 7).

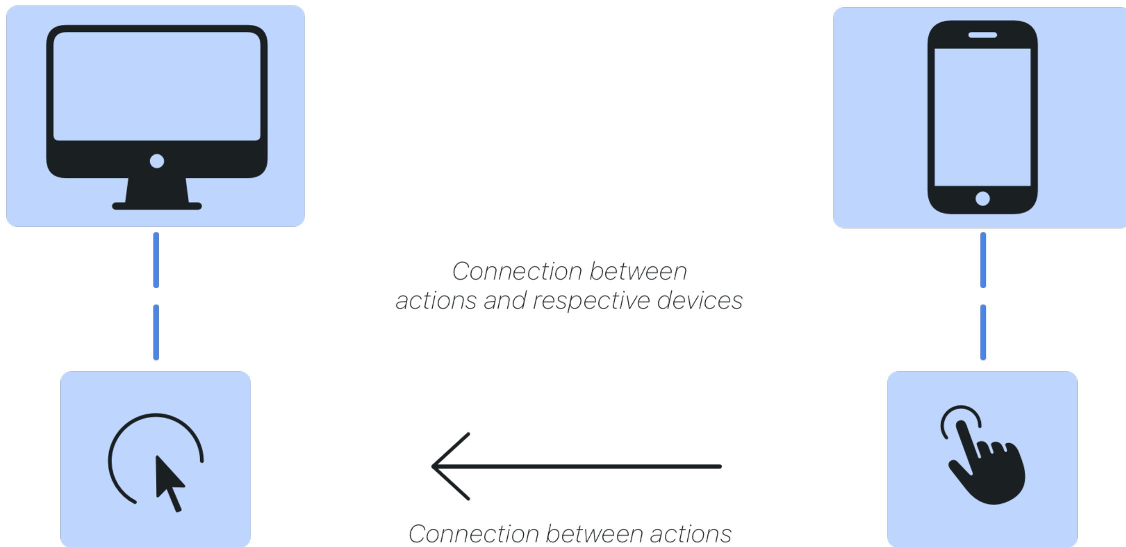

Figure 7: Two connection types present in a simple cross device interaction

The implementation of the first type of connection is discussed briefly in the last section. In this case, one of the two objects connected to each other is usually dependent on the other to function and interact with other devices. To represent this hierarchical connection model and close connections between these objects the toolkit uses the puzzle-based interface that allows users to connect the two objects (Figure 8).

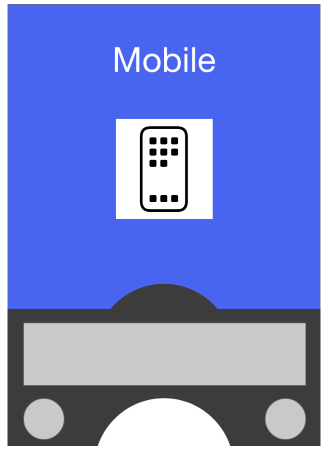

Figure 8: A Device block and interaction container connected together

connection using the puzzle-like interface

The connection between two different devices requires a metaphor that can represent the distribution of actions around the interactive space and the movement of data from one device to another. For this type of connection, the toolkit proposes two different connection interfaces (Figure 9). The first uses directional wires that slot into the connection point is the interaction containers. These wires provide a clear visual representation of the data travel between devices; however, they limit the interactions to being single input single output interactions. The toolkit also supports a second connection technique using colour-coded tokens that represent the inputs and outputs for each interaction in order to remove the limitation exposed by the first connection method. This allows for more complex interactions involving multiple inputs and outputs; however, it lacks the first connection method's visual representation.

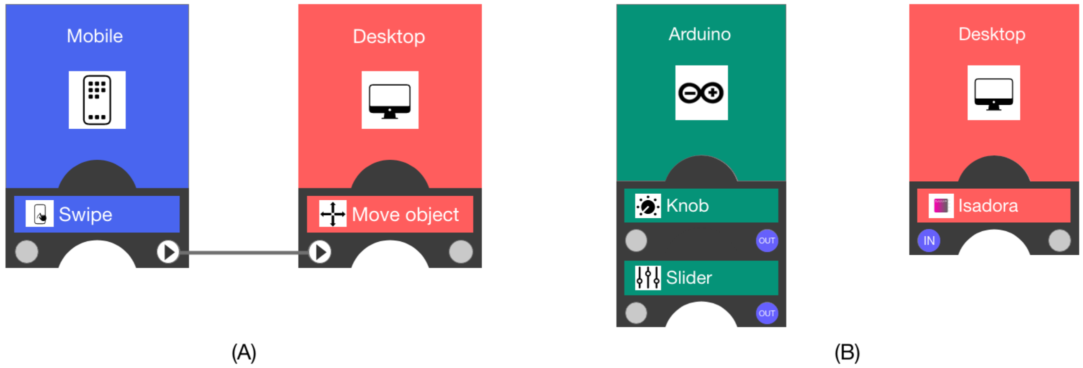

Figure 9: Two connection methods for connecting interactions (a) Connecting wire (b) Passive tokens

Graphical Interface

In order to complete the design process, the user can scan the layout of the interaction framework using a mobile device equipped with a camera. This scanned layout can be imported into the supporting graphical interface in order to complete the design process. The GUI offers an opportunity to get the user to input any additional information that might be needed to properly compile the interaction scenario, for instance, images, 3D models, Audio files etc.

Evaluation

An exploratory user study was designed to test the usability of the toolkit framework. The goal of the user study is to measure the impact of the toolkit in improving understanding and simplifying the design of cross-device experiences for the target users. Additionally, the study is used to collect feedback on some unfinished features of the toolkit such as the connection interfaces.

Study Design

In light of the COVID pandemic and the required social distancing the study was designed such that it could be performed virtually. In order to ensure that the study was accessible to the largest possible user base, the participants were provided templates for low fidelity paper prototypes and a mock-up of the graphical interface. This would ensure that the participants could experience and provide feedback on the design framework without access to the physical prototypes.

At the start of the study, participants are presented with a paper prototype (Appendix C) for the toolkit. Once they are familiar with the components of the toolkit, they are guided through the design process for creating 2 interaction scenarios:

Scenario 1: This scenario uses a simple one input to one output interaction in order to explain the flow of the toolkit to the participant and help them understand the basic working of the toolkit. Using two device cards, one Desktop and one Mobile, the participant adds interaction cards to each device and then connects them using the connecting wire method.

Scenario 2: The participant is introduced to connecting one output to multiple inputs and the token connection method. The interaction scenario uses an Arduino with a potentiometer and a slider as inputs for the user that is then passed into Isadora in order to produce outputs. This scenario is meant to be closer to a realistic interaction scenario. It also provides an opportunity to test the two connection methods.

Once the participants have completed the design walkthrough, they are asked to rank their experience with the toolkit on a 5-point Likert Scale. The questions are derived from the System Usability Scale (SUS) and are modified to be relevant to the toolkit design. Following the questionnaire, the participants are asked to provide subjective feedback on their experience with the toolkit as well as support for devices and interactions and the connection methods (Appendix D).

Pilot Study feedback

An informal pilot study was conducted with four (2M/2F; avg. age 25) participants in order to identify potential problems with the user study protocol as well as gather some preliminary feedback on the toolkit. The participants self-identified their area of expertise as design or development. The participants had varying degrees of experience with programming cross device interactions with three of the participants having less than 2 years of experience. All participants were provided with the prototypes for the toolkit and were guided through the process of creating the interaction scenarios verbally. This part of the study took about 20 mins in each case. All participants were easily able to identify the different parts of the toolkit. All of the participants seemed at ease with the toolkit once they had finished the first interaction scenario. Three out of the four participants preferred the wired connection method to the wireless one stating that it was easier to visualise the data flow and connection even though it might get complicated with a large number of interactions. Generally, the participants expressed that the toolkit was extremely easy to use, and it helped them visualize connections. Some of the responses from the participants were "Having the puzzle shape really helped to idealise out where they[interaction containers] go," “Tangible things help me process things a lot better” “The fact that it was a literal puzzle made it really easy to understand what was necessary to create interactions,” “ the colour coding really helped but the terminology needs to be simpler”.

There were some common concerns expressed by the participants about the toolkit as well. Three of the participants felt that they did not fully understand what the terminology used on interaction cards meant. They found it to be inconsistent as the terms ‘Rotate’ and ‘Isadora’ were too ambiguous and while the colours helped identify the proper places for the cards, the interaction behind the card was not clear. One participant also stated that they did not understand the need for such a tool when designing the first interaction however the second scenario helped clarify that aspect of the toolkit. One of the participants felt that they needed some confirmation as they were placing the interaction cards in order to ensure that the connection had been made.

The results from the study showed that the design of the toolkit was promising and helped improve the understanding of cross-device interaction design for novice users. It also highlighted the problems with terminology used in the toolkit and the need for an onboarding process in order to use the toolkit effectively.

User Study Limitations

The study was designed to be conducted in a virtual environment thus there were some gaps in the protocol. Even though the participants were provided with a template for the paper prototypes, some potential participants did not have access to colour printers which required them to back out of the study or create their own prototypes. The use of paper prototypes also restricted the ability to test the physical affordances offered by the tangible elements.

Next Steps

There have been numerous changes to the scope of the project over the last few months given the pandemic. The pilot study shows that it is possible to develop and use the toolkit in a remote setting. However, the study also highlighted some gaps in the study protocol as well as the toolkit design. The following section highlights the next steps for the toolkit in regard to the user study protocol as well as the future design and implementation of the toolkit.

Testing protocol

The pilot study brings to light a few changes that need to be made to the study protocol before conducting a formal evaluation of the toolkit design. While the study helped participants learn how to use the toolkit it lacked the real-world examples needed in order to visualise the practical use cases for complex interactions. The paper prototype was useful for testing design flow and the understanding however 3D tangible prototypes need to be used in order to test the physical affordances and graspability of the toolkit. The delivery of prototypes required for the study may need to be improved as some participants may not have easy access to printers.

Toolkit Design and Implementation

The pilot study provided some important preliminary feedback about the design of the toolkit. While the tangible objects may help in improving haptic feedback, they still lack the dynamic feedback needed to convey when an interaction is possible and when the user tries to create an incompatible interaction. The participants of the study expressed some concern regarding the terminology used on the interaction cards. These concerns can be split into 3 major problems; (1) The labels on the cards did not fully describe the action, (2) Some labels could be interpreted in multiple ways as the same action could be performed with both devices, (3) Some of the action cards did not represent actions but rather represented sensors or software.

While the first two problems need more work, the last problem can be fixed by understanding the action that the user is expecting and using appropriate language with respect to the same. This has already been done with the Isadora card (Figure 10) and other cards will need the same rework before the next round of user testing.

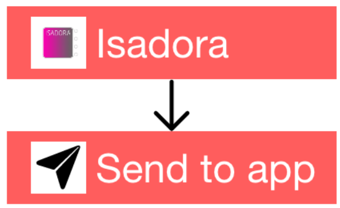

Figure 10: Change to the Isadora card

The physical aspects of the toolkit have a robust flow and the pilot study illustrates how it may help users visualise cross-device interaction design, however, the supporting virtual infrastructure needs to be developed in order to use the toolkit for practical applications. The toolkit could build upon existing universal cross device toolkits discussed previously and serve as a physical extension to these toolkits. This would ensure that the toolkit could be used by novice designers and allow them to seamlessly move over to a software-based toolkit when they are comfortable with developing cross-device connections.

Conclusion

The overarching aim of this project was to design a tangible cross-device authoring environment with low threshold, high ceilings and wide walls. More specifically this paper focused on the design framework for such a tool, providing a set of tangibles that could help users understand the connections and data flow responsible for cross-device interactions. Although the toolkit is still a work in progress, this project highlights that using Tangible User Interfaces could be a potential avenue for improving the understanding of cross-device interaction design among inexperienced users. The pilot user study illustrated that the toolkit framework helped designers better understand the connections behind cross-device interactions while still allowing them to focus on the high-level experience design. Overall, such a toolkit could not only be useful to simplify the design of cross-device interactions but may also improve the understanding of the design of interactive spaces as a whole.

Appendix A: Breakdown of Devices Used in Experience Design

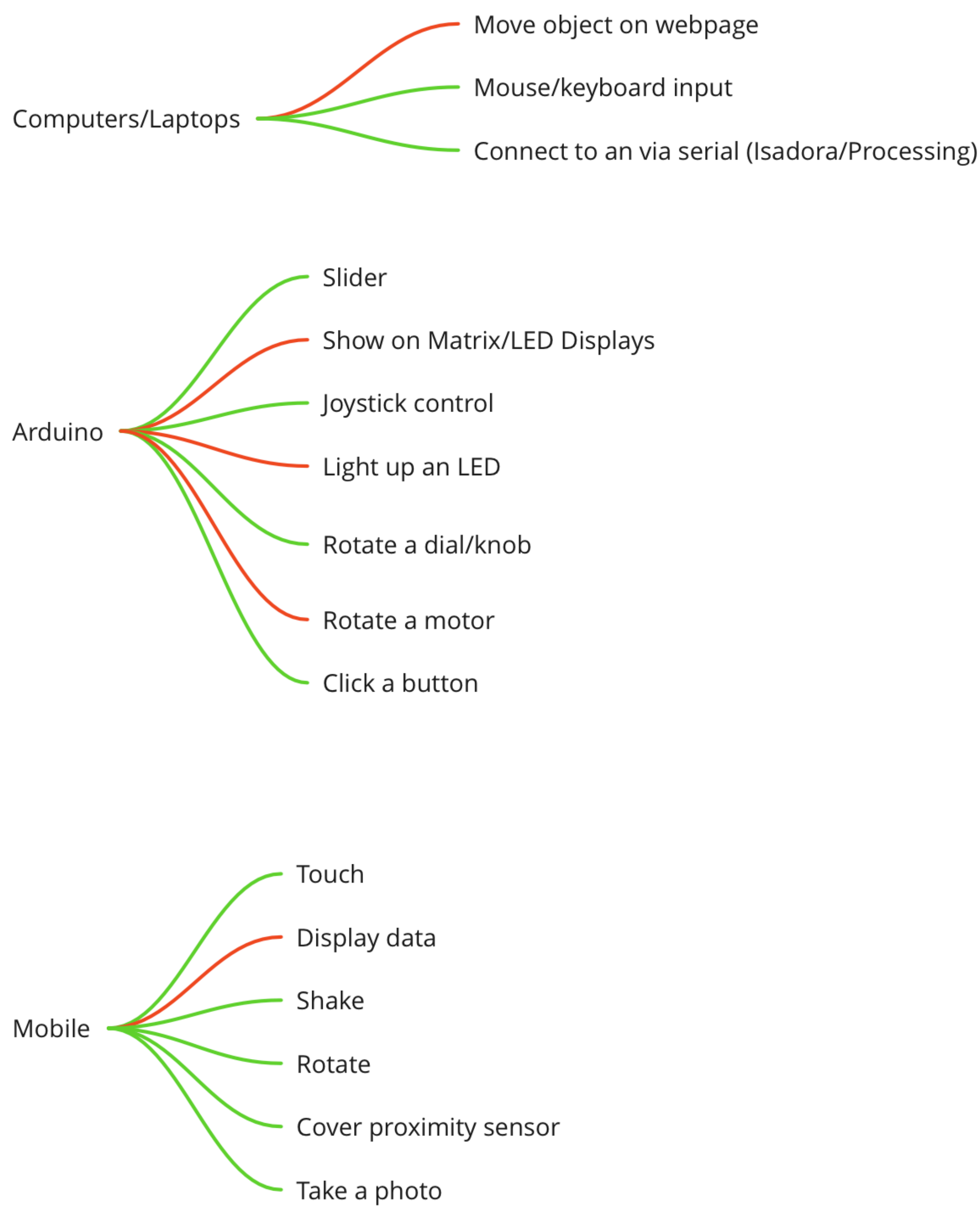

Figure A-1: Device used, and actions commonly performed in cross-device interactions

Appendix B: Toolkit Support V.1

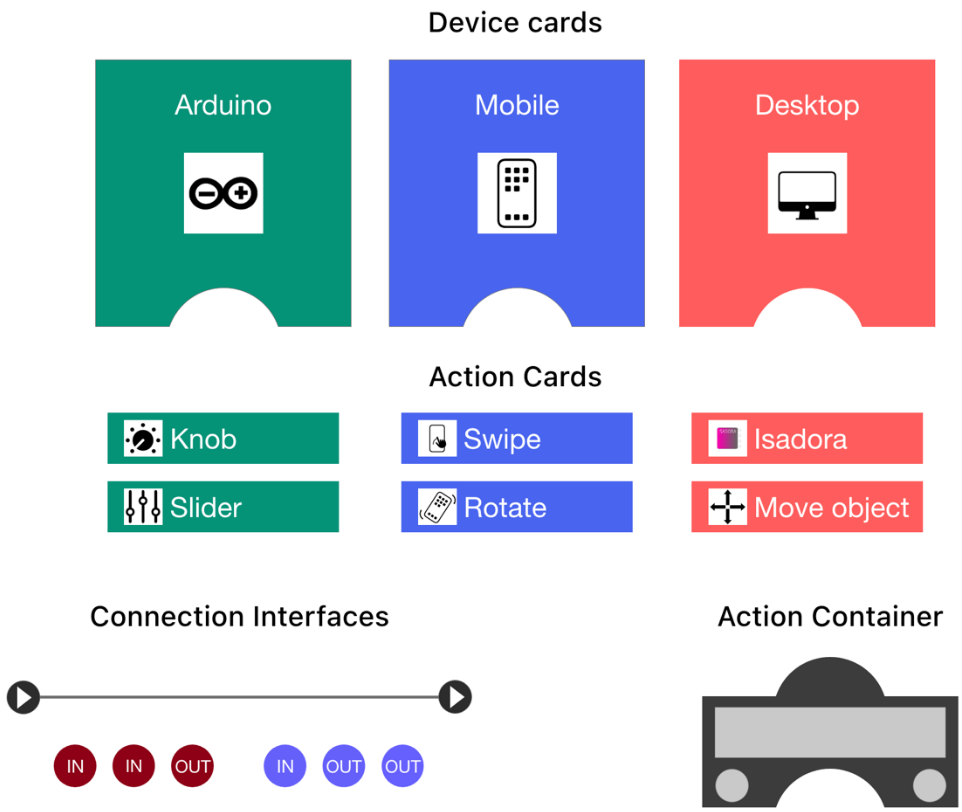

Figure B-1: Tangibles supported by the toolkit framework

Appendix C: Paper Prototype used for the User Study

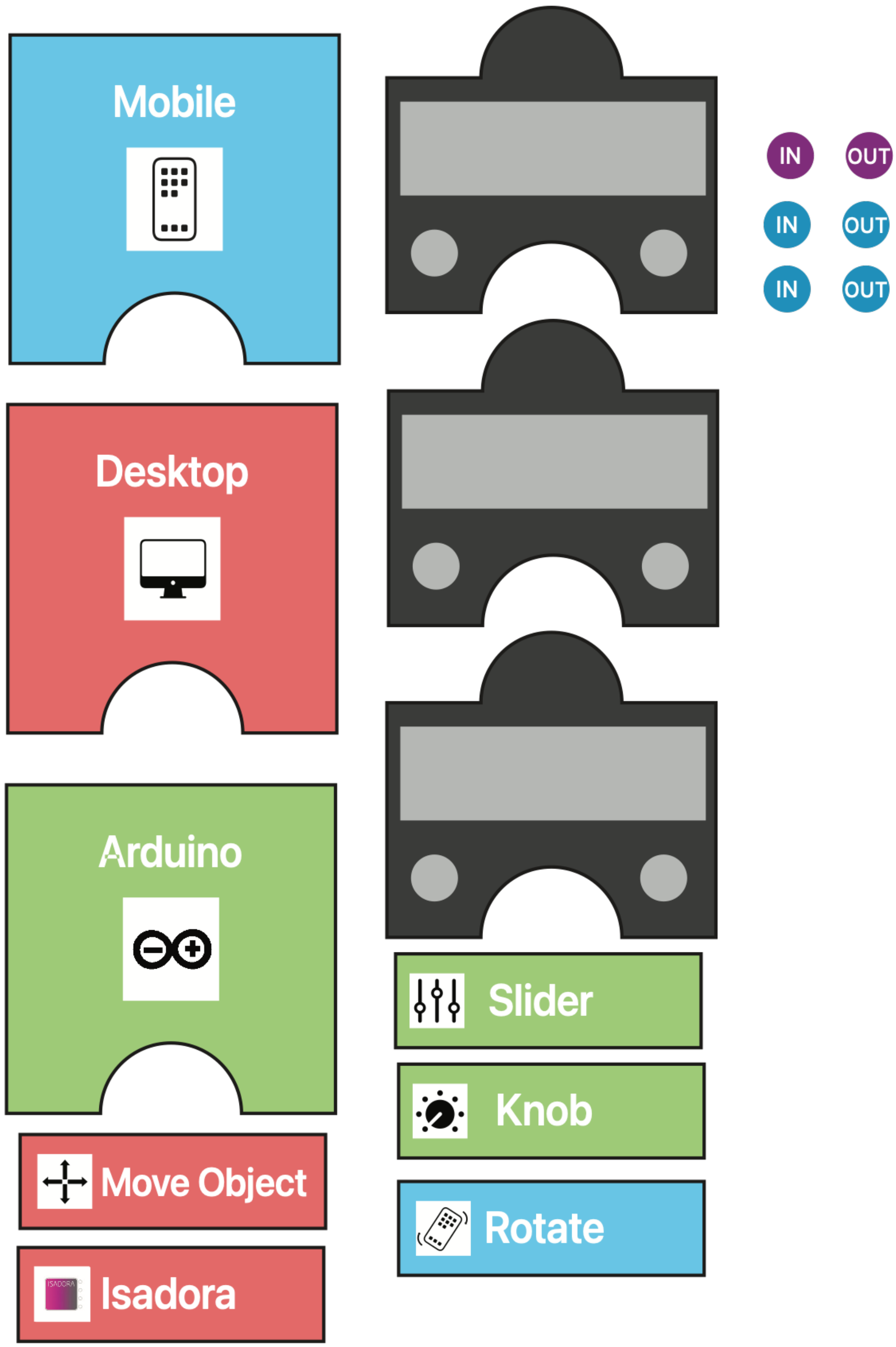

Figure C-1: User study prototype tested by participants

Appendix D: User Study Questionnaire

Objective Questionnaire:

Age: ______ years
Profession/Area of expertise: ___________________
Experience with programming: _____ years
Experience with multi-device development: _____ years
Experience with developing interactive experiences: ____ years

Answer the following questions with one of five responses that range from Strongly Agree to Strongly disagree:

I found the system unnecessarily complex.
I thought the system was easy to use.
I think that I would need the support of a technical person to be able to use this system.
I found the various functions in this system were well integrated.
I thought there was too much inconsistency in this system.
I would imagine that most people would learn to use this system very quickly.
I found the system very cumbersome to use.
I needed to learn a lot of things before I could get going with this system.

Subjective Interview Script:

Background questions:

1. Have you built an interactive experience before using multiple devices? Elaborate.
2. Do you have any experience building interactive experiences with sensor-based physical computing systems?
    a. If yes, then what systems and what was the interaction that they built?
    b. If no, what has been the barrier to not using such systems

Toolkit specific questions:

1. How comfortable would you be using this system?
    a. If not, what aspect of the toolkit did you find cumbersome?
2. The current version of the toolkit supports 3 devices and a couple of interactions per device. How did you find these interactions in terms of usability?
    a. What might be added to improve practical applications?
1. Two different connection methods were used in the toolkit. What are your thoughts about these connection methods?
    a. Which one do you prefer and why?

Bibliography

Acuna, K. (2014). *I Bought A $47 Interactive Harry Potter Wand And It Was Totally Worth It*. Business Insider. https://www.businessinsider.com/harry-potter-interactive-wand-review-2014-8

Andersson, D. (2015, January 27). Impressive Peter Pan's Flight interactive queue debuts dazzling pixie dust at Walt Disney World. *Inside the Magic*. https://insidethemagic.net/2015/01/impressive-peter-pans-flight-interactive-queue-debuts-dazzling-pixie-dust-at-walt-disney-world/

Billinghurst, M. (2003). No more wimps: Designing interfaces for the real world. *Proceedings of the 4th Annual Conference of the ACM Special Interest Group on Computer-Human Interaction - CHINZ '03*, 5–8. https://doi.org/10.1145/2331829.2331831

Block, F., Hammerman, J., Horn, M., Spiegel, A., Christiansen, J., Phillips, B., Diamond, J., Evans, E. M., & Shen, C. (2015). Fluid Grouping: Quantifying Group Engagement around Interactive Tabletop Exhibits in the Wild. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15*, 867–876. https://doi.org/10.1145/2702123.2702231

Bongers, B. (2002). Interactivating Spaces. *Proceedings of the Symposium on Systems Research in the Arts, Informatics and Cybernetics*. https://www.researchgate.net/profile/Bert_Bongers/publication/249730309_Interactivating_S_p_aces/links/0046352e5895f82b49000000/Interactivating-S-p-aces.pdf

Bouchard, D., & Daniels, S. (2015). Tiles that Talk: Tangible Templates for Networked Objects. *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction - TEI ’14*, 197–200. https://doi.org/10.1145/2677199.2680607

Bowell, A. (2020, July 22). *Bolt visual scripting is now included in all Unity plans*. https://blogs.unity3d.com/2020/07/22/bolt-visual-scripting-is-now-included-in-all-unity-plans/

Brudy, F., Holz, C., Rädle, R., Wu, C.-J., Houben, S., Klokmose, C. N., & Marquardt, N. (2019). Cross-Device Taxonomy: Survey, Opportunities and Challenges of Interactions Spanning Across Multiple Devices. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI ’19*, 1–28. https://doi.org/10.1145/3290605.3300792

Casteleyn, S., Rossi, G., & Winckler, M. (Eds.). (2014). *Web Engineering: 14th International Conference, ICWE 2014, Toulouse, France, July 1-4, 2014. Proceedings* (Vol. 8541). Springer International Publishing. https://doi.org/10.1007/978-3-319-08245-5

Chawla, K., Chiou, M., Sandes, A., & Blikstein, P. (2013). Dr. Wagon: A “stretchable” toolkit for tangible computer programming. *Proceedings of the 12th International Conference on Interaction Design and Children - IDC ’13*, 561–564. https://doi.org/10.1145/2485760.2485865

Chi, P.-Y. (Peggy), & Li, Y. (2015). Weave: Scripting Cross-Device Wearable Interaction. *Proceedings of the 33rd Annual ACM Conference on Human Factors in*

Computing Systems - CHI '15, 3923–3932. https://doi.org/10.1145/2702123.2702451

Ciolfi, L. (2004). *Situating "Place" in Interaction Design: Enhancing the User Experience in Interactive Environments*. 274.

Coutaz, J. (1987). PAC, an Object Oriented Model for Dialog Design. In H.-J. Bullinger & B. Shackel (Eds.), *Human–Computer Interaction–INTERACT '87* (pp. 431–436). North-Holland. https://doi.org/10.1016/B978-0-444-70304-0.50074-1

Di Geronimo, L., Husmann, M., & Norrie, M. C. (2016). Surveying personal device ecosystems with cross-device applications in mind. *Proceedings of the 5th ACM International Symposium on Pervasive Displays - PerDis '16*, 220–227. https://doi.org/10.1145/2914920.2915028

Hamilton, P., & Wigdor, D. J. (2014). Conductor: Enabling and understanding cross-device interaction. *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems - CHI '14*, 2773–2782. https://doi.org/10.1145/2556288.2557170

Harley, D., Chu, J. H., Kwan, J., & Mazalek, A. (2016). Towards a Framework for Tangible Narratives. *Proceedings of the TEI '16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction - TEI '16*, 62–69. https://doi.org/10.1145/2839462.2839471

Horn, M. S., & Jacob, R. J. K. (2007). Designing tangible programming languages for classroom use. *Proceedings of the 1st International Conference on Tangible and Embedded Interaction - TEI '07*, 159. https://doi.org/10.1145/1226969.1227003

Hornecker, E., & Buur, J. (2006). *Getting a Grip on Tangible Interaction: A Framework on Physical Space and Social Interaction*. 10.

Houben, S., Marquardt, N., Vermeulen, J., Klokmose, C., Schöning, J., Reiterer, H., & Holz, C. (2017). Opportunities and challenges for cross-device interactions in the wild. *Interactions*, *24*(5), 58–63. https://doi.org/10.1145/3121348

Ishii, H. (2008a). The tangible user interface and its evolution. *Communications of the ACM*, *51*(6), 32–36. https://doi.org/10.1145/1349026.1349034

Ishii, H. (2008b). Tangible bits: Beyond pixels. *Proceedings of the 2nd International Conference on Tangible and Embedded Interaction - TEI '08*, xv. https://doi.org/10.1145/1347390.1347392

Kaltenbrunner, M., & Bencina, R. (2007). reacTIVision: A computer-vision framework for table-based tangible interaction. *Proceedings of the 1st International Conference on Tangible and Embedded Interaction - TEI '07*, 69. https://doi.org/10.1145/1226969.1226983

Klum, S., Isenberg, P., Langner, R., Fekete, J.-D., & Dachselt, R. (2012). Stackables: Combining tangibles for faceted browsing. *Proceedings of the International Working Conference on Advanced Visual Interfaces - AVI '12*, 241. https://doi.org/10.1145/2254556.2254600

Lechelt, Z., Rogers, Y., Marquardt, N., & Shum, V. (2016). ConnectUs: A New Toolkit for Teaching about the Internet of Things. *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems - CHI EA '16*, 3711–3714. https://doi.org/10.1145/2851581.2890241

Marco, F. A., Penichet, V. M. R., & Gallud, J. A. (2011). Drag & Share: A Shared Workspace for Distributed Synchronous Collaboration. In J. A. Gallud, R. Tesoriero, & V. M. R. Penichet (Eds.), *Distributed User Interfaces* (pp. 125–132). Springer London. https://doi.org/10.1007/978-1-4471-2271-5_14

Marco, J., Cerezo, E., & Baldassarri, S. (2012). ToyVision: A toolkit for prototyping tabletop tangible games. *Proceedings of the 4th ACM SIGCHI Symposium on Engineering Interactive Computing Systems - EICS '12*, 71. https://doi.org/10.1145/2305484.2305498

Marquardt, N., Brudy, F., Liu, C., Bengler, B., & Holz, C. (2018). SurfaceConstellations: A Modular Hardware Platform for Ad-Hoc Reconfigurable Cross-Device Workspaces. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, 1–14. https://doi.org/10.1145/3173574.3173928

Matsuzawa, Y., Tanaka, Y., & Sakai, S. (2016). Measuring an Impact of Block-Based Language in Introductory Programming. In T. Brinda, N. Mavengere, I. Haukijärvi, C. Lewin, & D. Passey (Eds.), *Stakeholders and Information Technology in Education* (Vol. 493, pp. 16–25). Springer International Publishing. https://doi.org/10.1007/978-3-319-54687-2_2

Muender, T., Reinschluessel, A. V., Drewes, S., Wenig, D., Döring, T., & Malaka, R. (2019). Does It Feel Real?: Using Tangibles with Different Fidelities to Build and Explore Scenes in Virtual Reality. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19*, 1–12. https://doi.org/10.1145/3290605.3300903

Myers, B., Hudson, S. E., & Pausch, R. (2000). *Past, Present, and Future of User Interface Software Tools*. *7*(1), 26.

Nebeling, M., Teunissen, E., Husmann, M., & Norrie, M. C. (2014). XDKinect: Development framework for cross-device interaction using kinect. *Proceedings of the 2014 ACM SIGCHI Symposium on Engineering Interactive Computing Systems - EICS '14*, 65–74. https://doi.org/10.1145/2607023.2607024

Reas, C., & McWilliams, C. (2011). *Form+ Code: In design, art, and architecture*. Princeton Architecturel Press.

Resnick, M., Maloney, J., Hernández, A. M., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). *Scratch: Programming for Everyone*. 12.

Resnick, M., Myers, B., Nakakoji, K., Shneiderman, B., Pausch, R., Selker, T., & Eisenberg, M. (2005). *Design Principles for Tools to Support Creative Thinking*. 18.

Rubidge, S., & MacDonald, A. (2004). Sensuous Geographies: A multi-user interactive/ responsive installation. *Digital Creativity*, *15*(4), 245–252. https://doi.org/10.1080/1462626048520186

Scharf, F., Wolters, C., Herczeg, M., & Cassens, J. (2013). *Definition, Taxonomy and Applications*. 7.

Schreiner, M., Rädle, R., Jetter, H.-C., & Reiterer, H. (2015). Connichiwa: A Framework for Cross-Device Web Applications. *Proceedings of the 33rd Annual ACM*

Conference Extended Abstracts on Human Factors in Computing Systems, 2163–2168. https://doi.org/10.1145/2702613.2732909

Seyed, T., Azazi, A., Chan, E., Wang, Y., & Maurer, F. (2015). SoD-Toolkit: A Toolkit for Interactively Prototyping and Developing Multi-Sensor, Multi-Device Environments. *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces - ITS '15*, 171–180. https://doi.org/10.1145/2817721.2817750

Shaer, O., Leland, N., Calvillo-Gamez, EduardoH., & Jacob, RobertJ. K. (2004). The TAC paradigm: Specifying tangible user interfaces. *Personal and Ubiquitous Computing*, *8*(5). https://doi.org/10.1007/s00779-004-0298-3

Tarun, A. P., Mazalek, A., & Bellucci, A. (2016). *Prototyping "In The Wild" Interaction Scenarios With RE/Tk*.

TROIKATRONIX: ISADORA. (n.d.). Retrieved August 3, 2020, from https://troikatronix.com/

Ullmer, B., & Ishii, H. (2000). Emerging frameworks for tangible user interfaces. *IBM Systems Journal*, *39*(3.4), 915–931. https://doi.org/10.1147/sj.393.0915

Ullmer, Brygg, & Ishii, H. (1997). The metaDESK: Models and prototypes for tangible user interfaces. *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology - UIST '97*, 223–232. https://doi.org/10.1145/263407.263551

Ullmer, Brygg, Ishii, H., & Glas, D. (1998). mediaBlocks: Physical containers, transports, and controls for online media. *Proceedings of the 25th Annual*

Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '98, 379–386. https://doi.org/10.1145/280814.280940

Ullmer, Brygg, Ishii, H., & Jacob, R. J. K. (2005). Token+constraint systems for tangible interaction with digital information. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *12*(1), 81–118. https://doi.org/10.1145/1057237.1057242

Valdes, C., Eastman, D., Grote, C., Thatte, S., Shaer, O., Mazalek, A., Ullmer, B., & Konkel, M. K. (2014). Exploring the design space of gestural interaction with active tokens through user-defined gestures. *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems - CHI '14*, 4107–4116. https://doi.org/10.1145/2556288.2557373

Ventä-Olkkonen, L., Åkerman, P., Puikkonen, A., Colley, A., & Häkkilä, J. (2014). Touching the ice: In-the-wild study of an interactive icewall. *Proceedings of the 26th Australian Computer-Human Interaction Conference on Designing Futures the Future of Design - OzCHI '14*, 129–132. https://doi.org/10.1145/2686612.2686630

Weintrop, D., & Wilensky, U. (2015). To block or not to block, that is the question: Students' perceptions of blocks-based programming. *Proceedings of the 14th International Conference on Interaction Design and Children - IDC '15*, 199–208. https://doi.org/10.1145/2771839.2771860

Weiser, M. (1991). The Computer for the 21st Century. *SCIENTIFIC AMERICAN*, 12.

Wiethoff, A., Schneider, H., Küfner, J., Rohs, M., Butz, A., & Greenberg, S. (2013). Paperbox: A toolkit for exploring tangible interaction on interactive surfaces.

Proceedings of the 9th ACM Conference on Creativity & Cognition - C&C '13, 64. https://doi.org/10.1145/2466627.2466635

Wu, A., Mendenhall, S., Jog, J., Hoag, L. S., & Mazalek, A. (2012). A nested APi structure to simplify cross-device communication. *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction - TEI '12*, 225. https://doi.org/10.1145/2148131.2148180

Zigelbaum, J., Horn, M. S., Shaer, O., & Jacob, R. J. K. (2007). The tangible video editor: Collaborative video editing with active tokens. *Proceedings of the 1st International Conference on Tangible and Embedded Interaction - TEI '07*, 43. https://doi.org/10.1145/1226969.1226978

Zuckerman, O., & Gal-Oz, A. (2013). To TUI or not to TUI: Evaluating performance and preference in tangible vs. graphical user interfaces. *International Journal of Human-Computer Studies*, *71*(7–8), 803–820. https://doi.org/10.1016/j.ijhcs.2013.04.003