Families That Code Together Learn Together: Exploring family-oriented programming in early childhood with ScratchJr and KIBO Robotics

A thesis submitted by

Madhumita Govindarajan (Madhu Govind)

In partial fulfillment of the requirements of

Master of Arts in

Child Study and Human Development

Tufts University

May 2019

Committee Members

Marina Umaschi Bers, Ph.D. (Chair)

Eliot-Pearson Dept. of Child Study and Human Development, Tufts University

Christine McWayne, Ph.D.

Eliot-Pearson Dept. of Child Study and Human Development, Tufts University

Ricarose Roque, Ph.D.

Department of Information Science, University of Colorado Boulder

Abstract

Family-oriented programming has emerged as a fun, informal way for families to engage in creative activities using programming technologies that teach children how to code. There is a gap in understanding parent-child interactions using graphical versus tangible programming interfaces. This thesis aims to explore how children ages 5-7 and parents jointly program using the screen-based ScratchJr app or the tangible KIBO robotics kit, two playful coding technologies for early childhood. Utilizing a mixed-methods approach, this thesis seeks to identify the roles exhibited by families at community-based Family Coding Day events and explore the affordances of ScratchJr and KIBO for promoting these roles. Results showed that families' role engagement did not differ between ScratchJr and KIBO. Regardless of interface, children engaged highly as Planners and parents as Coaches. Qualitative findings suggest that family-oriented programming in early childhood parallels existing literature on joint media engagement. Implications and future directions are discussed.

Keywords: family programming, coding, parent-child interaction, graphical and tangible interfaces, informal learning, early childhood.

Acknowledgements

I entered the graduate program at Eliot-Pearson almost two years ago, bright-eyed and eager to gain as much knowledge and experience as I could during my time here. As I continue to learn and grow each day, I am grateful for the opportunity to be mentored by an incredible professor, newly appointed department chair, and woman– to name just a few of the many "hats" Dr. Marina Umaschi Bers wears. From her thoughtful feedback on previous drafts to the many informal conversations we've had in the most random of places, I am truly indebted to her dedication towards my success, and I look forward to continuing my doctoral work under her advisement. I am further grateful to the other members of my thesis committee, Dr. Christine McWayne and Dr. Ricarose Roque. Each of them provided unique insight and valuable feedback towards this thesis, and I have learned a great deal about methodology and analysis from our conversations. I thank them deeply for their time and commitment to serving on my committee.

This thesis most definitely would not be possible without the incredible efforts contributed by members of the DevTech Research Group. I wish to thank Dr. Amanda Sullivan and Kaitlyn Leidl for their initial efforts in piloting Family Coding Day events and bringing family engagement to the forefront of DevTech's research efforts. Thank you also to Emily Relkin, Anna Kasagawa, Kathryn Tweel, and Melissa Viezel for their time and support in coordinating and facilitating these events. My deepest gratitude goes to Hyejin Im and Cathy Cowell for their countless hours spent towards data collection and coding for the Parent-Child follow-up study, and to Anne Drescher for her thoughtful comments and moral support. Lastly, thank you to Emily Relkin, Amanda Strawhacker, and Dr. Ziva Hassenfeld for their friendship and mentorship these last two years. I have learned and continue to learn a great deal from each of you and have immensely appreciated our conversations that have led me to become a better researcher.

I would also like to acknowledge several individuals in the Eliot-Pearson department who I sought out for guidance at various points during the writing of this thesis. Thank you to Meera Menon for hosting monthly check-in meetings with the master's thesis students, Sunah Hyun for her guidance on qualitative and mixed-methods designs, and Dr. Mariah Contreras and Dr. Jayanthi Mistry for their additional expert guidance as I worked on various aspects of this thesis in their courses.

To the families who participated in this research and the individuals who contributed to recruiting and facilitating events for families (Dr. Hanna Gebretensae, Cora Carey, and others): thank you. This work would not be possible without you. And to my own family– Amma, Appa, Gowtham, Simba, and Neal– thank you for your endless support in each endeavor I choose to pursue. I hope to always make you proud.

This work was generously funded by the National Science Foundation (DRL-1118664) and the Scratch Foundation.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Unlike their parents in decades past, children growing up in the twenty-first century are exposed to a wide range of technological tools. From e-books to tablets where children can download games or video chat with family members across the world, technology has transformed the way children and families play, learn, and interact with one another. These new ways of engaging with technology, fueled by advancements in human-computer interaction research, have paved the way for new programming technologies that teach young children how to code and to become producers of their own creative artifacts (Bers, 2012; Resnick & Silverman, 2005; Yu & Roque, 2018).

Coding technologies, such as those that involve robotics or apps with programming languages, come at a time when the demand for computing jobs in the United States workforce is at an all-time high (Bureau of Labor Statistics, 2015), and researchers have highlighted the vast cognitive benefits of introducing computer science to young children (Bers, 2018; Clements & Gullo, 1984; Resnick & Silverman, 2005; Strawhacker & Bers, 2018). For instance, when children code they learn how to think in systematic ways and use their creativity to solve problems (Bers, 2018). These skills illustrate the phenomenon many scholars refer to as computational thinking, which they argue are a universal set of skills that should be introduced earlier in K-12 education (Grover & Pea, 2013; Wing, 2006; Wing & Stanzione, 2016). These advancements in the curriculum have led to policy changes at national and international levels, such as the adoption of K-12 computer science standards and frameworks (Code.org, 2018; Pretz, 2014). However, given that many schools have limitations on the amount of time students can spend using these new programming technologies, understanding the role of parents in facilitating children's engagement with these tools outside of school spaces is critical (Roque, Lin, & Liuzzi, 2016).

Family-oriented programming has thus emerged as a fun, informal way for parents and children to come together and jointly engage in programming activities. Initiatives such as Family Code Night and Family Creative Learning bring families together in informal settings such as afterschool programs or community centers to work on creative coding projects (Pearce & Borba, 2017; Roque, 2016). These projects, however, primarily focus on children aged seven and older. There is a current gap in understanding family-oriented programming in early childhood, a period of children's development where much of their learning occurs through play-based activities with their caregivers (Bers, 2018; NAEYC & Fred Rogers Center, 2012; Rideout, 2017).

Researchers can perhaps understand the phenomenon of family-oriented programming in early childhood by examining the literature on joint media engagement. Joint media engagement is defined in Takeuchi and Stevens (2011) as "spontaneous and designed experiences of people using media together… [such as through] viewing, playing, searching, reading, contributing, and creating, with either digital or traditional media" (p. 9). The literature on joint media engagement suggests that both parents and children benefit from engaging with technology together. For instance, parents may provide verbal, emotional, physical, or cognitive scaffolding support to help children understand difficult concepts or guide their learning. Children can also share their perspectives and even their expertise with their parents, which can serve as a valuable opportunity for parents and children to reverse traditional roles of teacher and learner (Barron, Martin, Takeuchi, & Fithian, 2009; Connell, Lauricella, & Wartella, 2015; Takeuchi & Stevens, 2011). These studies indicate that parents and children's roles during joint media engagement may be influenced by their varying levels of technological fluency and media experience.

However, different technological interfaces might promote or hinder distinctive types of joint engagement, as they invite varying kinds of interactions given unique design features (Yu & Roque, 2018). Some platforms involve the use of digital screens, such as a tablet or computer, whereas other platforms involve physical parts, such as wooden blocks or puzzle pieces. There are benefits and limitations to *graphical* and *tangible* programming interfaces, which impact how children interact and learn with these tools (Horn, Solovey, Crouser, & Jacob, 2009; Strawhacker, Sullivan, & Bers, 2013). For instance, some children prefer wooden blocks to screens because of the blocks' tangible nature, which can make tasks such as sequencing more concrete and thus easier to grasp for younger children (Horn & Bers, 2018). Conversely, other children may be already familiar with screen media and find graphical platforms more enticing (Horn et al., 2009; Sapounidis & Demetriadis, 2013). The unique characteristics of graphical and tangible interfaces might influence how children and parents engage in joint programming, during which the interface is shared among multiple individuals.

This thesis aims to explore how children ages 5-7 and their parents jointly program using the screen-based ScratchJr app or the tangible KIBO robotics kit, two developmentally appropriate, playful coding technologies for early childhood developed by the DevTech Research Group at Tufts University. ScratchJr, the graphical programming interface used in this work, is a tablet-based app in which children design their own characters and backgrounds and string together programming blocks to animate their characters (Bers & Resnick, 2015; Flannery et al., 2013). The tangible programming interface used in this work is the KIBO robotics kit, which is a robot that has an embedded scanner and can be programmed using a sequence of wooden blocks containing barcodes (Bers, 2018; Sullivan, Bers, & Mihm, 2017).

Understanding how ScratchJr and KIBO are utilized in formal learning settings only partly addresses young children's engagement with these tools. It is necessary to explore how parents engage with these playful learning technologies alongside their children in informal settings. Through the close examination of parent-child dyadic interactions with ScratchJr and KIBO, this thesis seeks to identify roles exhibited by parents and children during joint programming activities with graphical versus tangible interfaces, as well as explore the opportunities that each interface provides for parents and children to assume those roles.

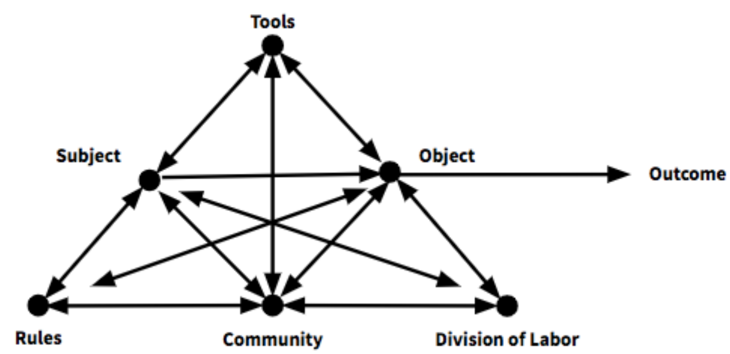
Chapter 2: Literature Review

There are two bodies of literature that predominately help to inform this work on family-oriented programming in early childhood—joint media engagement (JME) and human-computer interaction (HCI). The basis of JME is that young children do not necessarily engage with technology on their own; often involved in children's technological experiences are siblings, parents, caregivers, or similarly aged peers (Rideout, 2014, 2017; Takeuchi & Stevens, 2011). With new technologies that have transformed the way children play and interact with others, the field of HCI serves as an important discipline for understanding how people interact with technological tools and how users' experiences can vary depending on both contextual factors and the design of the tools themselves (Horn et al., 2009).

Vygotsky's social development theory emphasizes the role of social and cultural context in shaping development. He theorized that when children engaged in something too complex for their current level of understanding, they existed in this temporary "zone of proximal development", during which outside guidance from parents, teachers, or peers would allow them to attain that knowledge (Vygotsky, 1980). This theory inherently positions the child as novice and the parent as expert; however, with new technologies, it is likely that the parent is also learning alongside their child. Researchers have used the term "participatory learning" to characterize the role of parents as co-learners when they engage in play activities along with their children (Clark, 2011). This type of learning involves some level of dialogue between parents and children, such as when parents provide prompts or ask questions, and children step up as leaders to provide explanations and contribute actively to the conversation. The result of participatory learning is that parents and children engage collaboratively and actively (Clark, 2011).

The phenomenon of children jointly engaging with technological tools with their parents can be best understood through the activity theory model, which was developed by Leont'ev, a student of Vygotsky. Depicted in Figure 1, the second generation activity theory model by Engeström (1987) illustrates the inter-relationships between people, technological tools, and activities that require the use of the tools to achieve a particular outcome, and the organizational and societal factors that help define the nature of those activities (Kaptelinin & Nardi, 2018; Leont'ev, 1978). This framework proves useful to ground the research presented here by providing a visual model that brings JME and HCI together. When children and parents use new technologies together (i.e., JME), they develop rules about who does what and how they are going to share the technology-mediated experience together, which depends on how the technology is utilized and can be shared between them (i.e., HCI).

Figure 1. Second generation activity theory model (Engeström, 1987)



In this work, I use this activity theory model to apply to the context of family-oriented programming. Children (subject) engage with programming technologies (either ScratchJr, a graphical interface, or KIBO, a tangible interface) in order to participate in the activity of coding (object) and thus engage in collaborative coding and develop computational thinking skills (outcome). When parents (community) co-engage with their children in programming activities, they may assume or develop "roles" such as teacher or observer, which are made explicit through their collaboration and sharing practices (rules) and how the parent and child choose to divide project tasks (division of labor). In this thesis I focus specifically on the "roles" exhibited by parents and children during family-oriented programming, zooming in closely on the rules and division of labor that may characterize parent-child interactions with different interfaces.

What types of parent and child roles have been found in existing models of family-oriented programming? Are there demographic or other factors that might influence the parent or child to take on specific roles during these events? Answers to these questions may be found in this next section, which presents the current literature on family-oriented programming events, most of which focus on children aged seven and older.

Family-Oriented Programming

As coding and creative computing platforms for children have become more popular, various models of family-oriented coding events have emerged, in which children and families are invited to create projects or play with coding software together. Family Code Night, for instance, provides free event kits for schools around the country to host large-scale family coding events for K-5 students and families using Code.org and unplugged activities (Pearce & Borba, 2017). The Family Creative Learning model consists of a series of workshops for school-age children and their family members to learn about and create projects using the Makey Makey invention kit and Scratch programming language (Roque, 2016; Roque, Lin, & Liuzzi, 2014, 2016). The Be A Scientist Family Science Program brings children and parents together for five-week workshops to engage in hands-on science and engineering-related activities (Pierson, Momoh, & Hupert, 2015).

Findings from family-oriented programming events indicate that these events are enjoyable for both parents and children and stimulate their interest in project-based tasks and creative problem-solving (Banerjee et al., 2018; Bers, 2007; Hart, 2010; Lin & Liu, 2012). For example, one study found that when parents and children jointly engaged with the LEGO-LOGO programming environment (a precursor of the well-known LEGO Mindstorms), parents felt more connected to their children as they listened to their ideas and learned from one another (Armon, 1997). Furthermore, rather than using trial-and-error methods to troubleshoot problems, children tended to be more purposeful and systematic with their programs (Lin & Liu, 2012) and performed significantly better on performance tasks when working with their parent (Hughes & Greenhough, 1995).

The research also points to several key factors that distinguish families' experiences at these events: parents' background in a STEM-related field, and children's prior experience with the tool due to classroom usage or extracurricular involvements. However, the literature is inconsistent. In some cases, parents with little to no background in programming or technology tended to allow their child to be the "driver" while they took on more passive roles, though some parents found it challenging to take on the role of novice (Lin & Liu, 2012; Roque et al., 2016). In other cases, parents with an information technology (IT) background showed higher competence and confidence when working with their children. However, these parents were also less likely to learn from their mistakes and to let children explore and tinker (Beals & Bers, 2006; Bers, 2007; Bers, New, & Boudreau, 2004; Feng, Lin, & Liu, 2011). Children who had prior experience with the tool enjoyed the unique opportunity to share their knowledge with their parents. Parents were typically excited about this role reversal, but at times felt that incorporating their own interests into the project was a challenge (Roque et al., 2014, 2016).

This body of work informs the research presented in this thesis by highlighting several factors that may influence how young children and families engage in joint programming activities. However, the kinds of interactions between parents and children might be different according to the kind of technological interfaces they have access to. The next section will identify the similarities and differences between graphical and tangible programming interfaces.

Graphical and Tangible Programming Interfaces

The literature on human-computer interaction suggests that different types of technological interfaces offer different types of user experiences. Graphical user interfaces (GUIs) utilize some sort of screen-based platform with visual elements such as icons, images, and windows (Sapounidis & Demetriadis, 2013; Strawhacker & Bers, 2015). Tangible user interfaces (TUIs), as defined by Horn and Bers (2018), "describe a class of computer interfaces that employ physical objects and surfaces as a means to both manipulate and represent digital information" (p. 662). One of the most salient differences between GUIs and TUIs is that when children engage with GUIs, they undergo mental operations to connect the act of clicking and dragging icons to the on-screen actions that they produce. Conversely, TUIs allow children to connect their digital environment to the physical world; by manipulating tangible objects, children can concretely represent the actions produced by the computer or robot (Sapounidis & Demetriadis, 2013).

Various studies have explored the similarities and differences in children's engagement with GUIs versus TUIs (Horn et al., 2009; Sapounidis & Demetriadis, 2013; Strawhacker & Bers, 2015; Strawhacker et al., 2013; Xie, Antle, & Motamedi, 2008). Younger children tended to find tangible interfaces more enjoyable and accessible, whereas older children who had more experience with computers found graphical interfaces easier to use (Sapounidis & Demetriadis, 2013). Key factors that impacted these findings included children's prior experience with similar interfaces and the spatial arrangement between the user(s) and the tool (Sapounidis & Demetriadis, 2013; Xie et al., 2008). Furthermore, regarding how children co-engage with GUIs and TUIs, almost all studies highlighted tangible interfaces as better platforms for sharing. Shaer and Hornecker (2010) discuss the benefit of multiple access points with TUIs, which means there are multiple objects or required actions that invite other play partners, such as peers or parents, to successfully co-engage in the activity. This advantage of TUIs is further supported by another study, which found that museum visitors were more likely to engage in collaborative behaviors with family members with the tangible exhibit than the screen-based exhibit (Horn, Crouser, & Bers, 2012). These findings suggest that perhaps tangible interfaces are better suited for family-oriented programming in early childhood, although this hypothesis has not been tested in the current literature.

Although there is much unknown about how young children and parents interact using different types of programming technologies, there is extensive literature on joint media engagement and how young children and parents interact using other types of technological tools. These next sections describe the roles identified in the literature on children and parents engaging together with different types of media and technology and the characteristics of those parent-child interactions, which may prove useful to understanding parent-child interactions during joint programming activities.

Roles of Parents and Children in Technological Experiences

Barron and colleagues (2009) identified seven different roles that parents may engage in to promote children's development of technological fluency: teacher, collaborator, resource provider, learning broker, non-technical consultant, employer, and learner. The researchers interviewed eight middle school students who were identified as highly technologically fluent, as well as their parents who had varying levels of technological expertise. The study findings indicated that parents can serve in a variety of roles to support their children's development of new media skills. Roles such as teacher or learning broker allow parents to share their knowledge with children, whereas parent roles such as employer or learner illustrate how children can be viewed as technological experts (Barron et al., 2009). In this study, however, at least one parent in each family worked in a STEM (science, technology, engineering, and mathematics) related field, and the children were young teenagers who inherently had a wider range of access to technology, so it is unclear how these roles might differ in families without prior STEM backgrounds and who have young children. This thesis sets to explore these questions.

Other studies have explored parent-child dynamics in informal museum settings and afterschool programs and found that with young children, parents typically served as facilitators of their children's learning, taking on teacher, coach, and playmate roles (Cheng, 2017; Sanford, Knutson, & Crowley, 2007; Swartz & Crowley, 2004). For example, one study by Swartz and Crowley (2004) identified five parental behaviors when they visited interactive museum exhibits with their children: observing children's interactions with the exhibit, encouraging children's play through words and gestures, directing their attention to different stimuli, describing features of the exhibit, and providing explanations for how those features worked. Furthermore, these aforementioned studies explored how adults and children engaged in physical spaces (e.g., visiting museums) versus digital spaces (e.g., looking up similar exhibit information online), finding that web-based experiences offered fewer opportunities for collaboration and social interaction. This finding brings into question how children and parents interact with and without technology-rich tools, which is described in the next section. This literature may prove useful to examining the varying affordances of graphical and tangible interfaces on family role engagement.

Parent-Child Interactions Using Different Tools

Prior work on joint media engagement has examined parent-child interactions in the context of e-books and tablet-based apps, finding contradictory results on the benefits and limitations of graphical tools versus other non-technological tools. Previous research has compared the social and cognitive benefits of e-books versus traditional books (Korat & Or, 2010; Krcmar & Cingel, 2014; Parish-Morris, Mahajan, Hirsh-Pasek, Golinkoff, & Collins, 2013). Some studies found that parents and children showed more active involvement and spent more time engaging with e-books, probably because of their novelty and the excitement of sound effects and animations. However, the buttons and gimmicks can be more distracting for parents and children to engage in high-quality dialogic reading, in which parents encourage children's active reading involvement by asking questions and relating the story to their personal experiences. This thesis sets to explore whether these dynamics are present when using graphical programming interfaces.

Other research studies have explored the differences between tablet-based apps versus traditional toys (Griffith & Arnold, 2018; Hiniker et al., 2018). For example, Hiniker and colleagues (2018) observed parent-child dyads as they interacted with various toys and tablet-based apps. They consistently found that when playing with toys such as bricks and puzzle pieces, children and parents assembled their play space so that each person had equal access to the parts. This arrangement helped facilitate conversation and allowed parents to take a more active role during play. Conversely, when playing with tablets, children and parents found it more difficult to share the screen and maintain dialogue, which led parents to take on bystander or spectator roles during tablet play (Hiniker et al., 2018). In another study conducted by Griffith and Arnold (2018), parents and children were observed using various learning tools: a traditional print book, a traditional math toy, a preliteracy app, and a math app. Similarly, the researchers found that children were more likely to lead the app interactions, whereas parents took on a more supportive role. Furthermore, children engaged with both traditional and app activities equally when parents were also highly engaged, exhibited playful behaviors and supported children's autonomy (Griffith & Arnold, 2018). These findings lend support for the three primary drivers of joint media engagement: spatial arrangement of the tool between the parent and child, the child's desire to engage with the tool, and the parent's desire to promote children's learning (Joan Ganz Cooney Center, 2014).

Because the literature on family-oriented programming is limited, understanding the factors that drive joint media engagement prove useful to this thesis. Factors such as spatial arrangement, prior STEM background, or experience with the tool may be key predictors of how parents and children assume certain roles during joint programming activities. Furthermore, this thesis seeks to explore the affordances of graphical and tangible interfaces on young children and parents' role engagement. The advancements in human-computer interaction research have led to a rise of technologically-rich graphical and tangible programming tools that allow children to not only interact with technology but also create with them (Bers, 2018; Resnick & Silverman, 2005; Yu & Roque, 2018). This next section describes the ScratchJr app and KIBO robotics kit, the respective graphical and tangible programming tools utilized in this thesis to explore family-oriented programming in early childhood.

Technologies Used in this Study

Both the ScratchJr app and KIBO robotics kit were developed by the DevTech Research Group at Tufts University through over a decade of research on creating developmentally appropriate, playful coding platforms for early childhood (Bers, 2018). ScratchJr and KIBO are block-based programming languages that were developed using a "learn by creating" or constructionist approach to children's engagement with technology (Ackermann, 2001; Bers, 2018; Papert, 1980). Studies of children using ScratchJr and KIBO show that children as young as four can grasp foundational computational thinking skills such as sequencing and debugging and can use their creativity to make their stories and ideas come to life (Bers, 2018; Kazakoff, Sullivan, & Bers, 2013; Strawhacker & Bers, 2015). Because ScratchJr and KIBO are open-ended coding platforms, they serve as the ideal platforms to explore family-oriented programming in early childhood and to compare how families engage with graphical versus tangible interfaces.

ScratchJr. ScratchJr is a free programming app designed with a "low floor, high ceiling, wide walls" approach so that individuals of diverse levels of experience can tinker with the graphical programming blocks to create imaginative stories and games (Bers, 2018; Flannery et al., 2013; Portelance, Strawhacker, & Bers, 2016; Resnick & Silverman, 2005). The app was designed as a collaboration among the DevTech Research Group at Tufts University, MIT Media Lab, and the Playful Invention Company and funded by the National Science Foundation and Scratch Foundation. Figure 2 shows a screenshot of the ScratchJr interface. Children drag and drop blocks into the programming area at the bottom of the screen and snap them together to create their characters' codes.

Figure 2. ScratchJr interface

Although ScratchJr is designed to be developmentally appropriate for children ages 5-7, its design features allow for parents and children to jointly engage with the interface and to co-create projects. There are over 20 different graphical programming blocks for children to use to program their characters; while some blocks, such as the "Start on Green Flag" and "Move Right" blocks are easy for children to grasp, other blocks such as "Send Message" and "Stop" require deeper understandings of programming languages, which adults may be able to unpack with children. Furthermore, ScratchJr users can customize their characters, backgrounds, and add up to four pages in their programs. Having all of these options can be quite enjoyable for children, but parents can help facilitate their experience by helping children narrow down their choices and engage in goal-oriented programming. Lastly, ScratchJr project sharing via Airdrop or email was a design feature to specifically promote adult engagement. In order to share projects, users must answer a double-digit calculation before being presented with the sharing options. This feature encourages adults, primarily educators and parents, to view children's projects and to share with members of their community (Bers & Resnick, 2015; Flannery et al., 2013; Portelance et al., 2016).

KIBO Robotics. The KIBO robotics kit was designed by Prof. Marina Umaschi Bers and the DevTech Research Group with a grant from the National Science Foundation to research and create a screen-free, tangible robotics platform that is developmentally appropriate for young children (Bers, 2018; Sullivan, Bers, & Mihm, 2017). The KIBO robot uses an embedded scanner to scan a series of wooden programming blocks containing barcodes. Figure 3 displays the KIBO robot with attached modules and sensors, wooden programming blocks, and accessories, such as the art platforms and the whiteboard expression module.

Figure 3. KIBO robotics kit

In order to program the KIBO robot, children can choose from 21 different programming blocks, each containing a specific icon that allows children to understand the function of the block without necessarily being able to read. Blocks are also color-coded to indicate function; for example, blue blocks allow KIBO to move, whereas yellow blocks allow KIBO's lightbulb module to light up in different colors. The programming blocks are also labeled with text, so that adults and children who are literate can identify the block names. There are also other design features of the KIBO robotics kit that invite children to co-play with peers and adults. For instance, users can collaboratively scan a program onto KIBO by taking turns with scanning and covering the other barcodes and work together to decorate the art platforms using arts and crafts materials. When programming with KIBO and sharing their projects with others, children engage in positive behaviors such as communication, collaboration, and community building (Bers, 2018; Elkin, Sullivan, & Bers, 2016; Sullivan, Bers, & Mihm, 2017).

ScratchJr versus KIBO. Prior studies with ScratchJr and KIBO (or early KIBO prototypes that used the CHERP programming language) indicate that there may be subtle differences in the way young children engage with these platforms. For instance, researchers noted differences in the classroom environment when children engaged with ScratchJr versus KIBO. There was generally more commotion in the tangible group with children moving around with their robots and assembling their block programs, whereas children were seated and focused more on their individual tablets in the graphical condition (Pugnali, Sullivan, & Bers, 2017). These findings bring into question the kinds of interactions children would have with their parents when they co-engage with ScratchJr versus KIBO and the types of affordances these tools provide for parents and children to assume different roles.

Chapter 3: Research Design

Background of Family Coding Days

In an effort to explore how children engaged with ScratchJr and KIBO in informal learning settings, the DevTech Research Group at Tufts University piloted "Family Coding Days" at the Tufts Eliot-Pearson Children's School and local Boston museums. Children between five and seven years old, as well as any family members ranging from grandparents to siblings, were invited to attend these family-oriented programming events involving ScratchJr or KIBO. Using feedback from families' experiences and the DevTech team's preparation for these events, a detailed protocol was devised for hosting a ScratchJr or KIBO Family Day, which was made freely accessible to anyone interested in facilitating a family coding event in their respective community (now available at http://sites.tufts.edu/devtech/learn-with-us/for-children-and-families/). The protocol, as well as all recruitment materials, consent forms, and surveys, were approved by the Tufts University Institutional Review Board (protocol #1612026).

Procedure. ScratchJr and KIBO Family Days were hosted by both the DevTech team and outside facilitators, who were recruited via the ScratchJr, KinderLab Robotics (the company that commercially sells KIBO), DevTech, and Family Code Night e-lists and social media platforms, all of which attracted a total readership of roughly 30,000 individuals. Interested facilitators completed a Google Form and were contacted by a DevTech researcher with the protocol. Detailed in the Family Day protocols were the following set of resources: family recruitment strategies, facilitator script explaining the purpose of the event, preand post-survey links, list of materials, sample agendas and activity prompts, parent tip sheets, suggested questions for project feedback, and off-screen activities such as coloring sheets and mazes. Facilitator checklists and flyer templates were additional resources provided to outside facilitators to plan for their events.

The protocol outlined three sample agendas (1 hour, 1.5 hours, and 2 hours) to allow facilitators to adapt the event to their community's needs. Each agenda contained similar activities: arrival and check-in, introduction to the technology, joint family coding time, "share and pair" feedback on projects, community sharing of projects, and closing. During the introduction portion, children separately engaged in off-screen games related to the technology while parents completed the pre-survey and received a step-by-step tutorial. During the joint family coding time, families were provided with three sample prompts (e.g., program a ScratchJr character/KIBO robot to perform a dance, be an animal, or act out a scene from a favorite book or movie) but were encouraged to use their own project ideas. After the joint coding session with some time allotted for peer feedback, parents completed a post-survey and joined their children for the community sharing of projects. Figure 4 details the three main components of Family Coding Days: *learn* about the technology, *create* a collaborative coding project, and *share* project with peers.

Figure 4. Basic Agenda of Family Coding Days

Quantitative data were collected in the form of pre- and post-surveys completed by parents via Qualtrics, an online survey engine. The 16-item pre-survey asked questions related to family demographics and children and parents' prior experience and interest in coding. At the end of Family Day events, parents completed a 13-item post-survey, which asked questions about their family's coding experience. Parents' pre- and post-surveys included open-ended and Likert-type scale questions, which are described in more detail in subsequent sections.

Participants. Between fall 2017 and summer 2018, 109 participants attended 14 ScratchJr or KIBO Family Coding Day events. The nine ScratchJr Family Day events (n = 58 families, 60 children; M_{age} = 6.4 years old) were hosted by both DevTech researchers and outside facilitators. Five KIBO Family Day events (n = 51 families, 57 children; M_{age} = 6.2 years old) were conducted solely by DevTech researchers. Seven families attended both ScratchJr and KIBO Family Day events and were removed from subsequent analyses in order to explore differences between independent samples. The remaining 95 participants' demographics are displayed in Table 1. Of note, Family Coding Day attendees included mostly highly educated mothers with children ages 5-7. Sixty-two percent of families did not have any prior experience with the technology before attending the event, and 32% of families had children who had some exposure to the technology through school.

Table 1. Family Coding Day Participant Demographics

Demographics	**ScratchJr** (*n* = 52)	**KIBO** (*n* = 43)	**Total** (*N* = 95)
Child Age			
5 years	8	10	18
6 years	17	7	24
7 years	13	8	21
Out of 5-7 range	11	13	24
Parent Gender			
Male	12	11	23
Female	40	32	72
Parent Education			
High school degree or	1	1	2
equivalent	0	2	2
Some college, no degree	4	3	7
Associate degree	15	14	29
Bachelor's degree	18	12	30
Master's degree	11	7	18
Professional degree			
Parent in STEM Profession			
Yes	22	11	33
No	27	27	54
Prior Experience with ScratchJr/KIBO			
Yes, child only	23	7	30
Yes, adult only	2	1	3
Yes, both child and adult	3	0	3
No, neither child nor adult	24	35	59

Research Constructs

In this thesis, I performed secondary data analysis on the Family Coding Days study conducted by DevTech researchers to explore differences in child and parent role engagement during ScratchJr versus KIBO Family Days. The types of roles children and parents could engage in during Family Coding Days were identified by looking at the related literature on parental roles in children's experiences with technology in informal settings. Through the process of examining the literature and assessing their relevance to the act of programming, there were five broad role categories that emerged: Planner, Observer, Teacher, Coach, and Playmate. Table 2 details the five role definitions and the relevant literature from which these role categories were created.

In the Family Day post-survey, parents were asked to rate the extent to which they engaged in each of these five roles on a Likert-type scale of 1 (not at all) to 5 (engaged frequently) with a brief description provided in each role (see Table 2). Likert-type scales were used instead of dichotomous variables (e.g., I did/did not engage as a Planner) in order to capture the magnitude of parents' perceived role engagement. Parents also reported the extent to which their children engaged in each of these five roles on the same 1-5 Likert-type scale. Findings from ScratchJr Family Day events indicated that although children and parents assumed multiple roles during their coding experience, there were significant differences in the roles reportedly assumed by children versus parents. In particular, children were reported to have engaged highly as Planners, whereas parents engaged highly as Coaches and Observers (Govind, Relkin, & Bers, manuscript submitted for publication). Similar findings were revealed in KIBO

Family Days, with children engaging highly as Planners and Playmates and parents engaging highly as Coaches (Relkin, Govind, & Bers, manuscript in preparation).

Table 2. Child and Parent Roles in Informal Learning Environments

Roles Explored in Family Coding Days	Related Roles from Literature
Planner: plans out project topic and delegates tasks to members of the group	***Creative designer:*** person initiates ideas and takes an active role in working with the tool (Roque et al., 2016)
Observer: lets others guide project creation, does not actively contribute to the group's coding activities	***Observer:*** person does not interact with others directly so that others can explore independently (Swartz & Crowley, 2004); ***Bystander or spectator:*** person watches others interact with the tool (Hiniker et al., 2018)
Teacher: explains some of the coding topics to the group during the activity	***Teacher:*** person possesses more knowledge about the subject and instructs others (Barron et al., 2009); person describes or explains topics (Sanford et al., 2007); person connects activity to other learning domains (Swartz & Crowley, 2004)
Coach: encourages and supports the group, offered suggestions to group members during the activity	***Nontechnical consultant:*** person provides encouragement or advice (Barron et al., 2009); ***Coach:*** person encourages, supports, and/or offers suggestions (Griffith & Arnold, 2018; Sanford et al., 2007)
Playmate: shares the fun, enjoyable parts of the activity with the group	***Collaborator:*** person shares the learning experience with others (Barron et al., 2009); ***Playmate:*** person shares fun, enjoyable aspects of activity without focusing on learning outcomes (Sanford et al., 2007; Swartz & Crowley, 2004)

Research Questions

From the preliminary analyses conducted separately with ScratchJr and KIBO Family Days, it was clear there were similarities and differences in how parents reported their role engagement, as well as how parents perceived their children's experiences. Thus, my first research question is **(1) What roles are exhibited when children and parents jointly program using ScratchJr, a graphical coding interface, versus KIBO, a tangible coding interface?** I was curious to explore whether family demographics or prior experience with ScratchJr or KIBO had an impact on their role engagement during Family Coding Day events. In addition, what did these roles look like in terms of children and parents' behaviors and actions? Can we document qualitative examples of how children engaged as Planners, or how parents engaged as Coaches? To answer these research questions, I observed individual parent-child dyads interacting with ScratchJr and KIBO, which served to illuminate the parent-reported findings from Family Coding Day events.

How children and parents develop and assume these roles might be dependent on the specific features of the ScratchJr and KIBO coding platforms. ScratchJr is a screen-based platform that requires a child and parent to share a single tablet, whereas KIBO is a screen-free robotics kit with many tangible pieces. These differences prompted my curiosity to explore how these interfaces impacted children and parents' role engagement. Thus, my second research question is **(2) What kinds of opportunities do ScratchJr and KIBO provide for family-oriented programming activities?** In other words, are there specific features of ScratchJr or KIBO that enabled parents or children to take on particular roles? To answer these questions, I looked for qualitative examples of how the design features of ScratchJr and KIBO enabled the parent-child dyads to share the technology-mediated experience together and how they chose to divide up tasks for their coding project.

Hypotheses. Although ScratchJr and KIBO are both open-ended coding platforms that teach young children how to code and produce creative projects, they are very different types of interfaces. As the human-computer interaction research suggests, tangible interfaces offer different types of user experiences than graphical interfaces. In reference to the first research question, I hypothesized that **(1a) families will assume different roles when jointly engaging with ScratchJr versus KIBO** and **(1b) children will assume different roles than parents during programming activities in general**. Regarding the second research question, prior literature suggests that children exhibit more sharing and collaborative behaviors when engaging with KIBO versus ScratchJr (Pugnali et al., 2017; Strawhacker & Bers, 2015; Strawhacker et al., 2013). Considering that KIBO contains more tangible parts which parents can easily access and thus facilitate children's engagement, my hypothesis was that **(2a) ScratchJr and KIBO provide different opportunities during family-oriented programming, with KIBO providing greater points of access**.

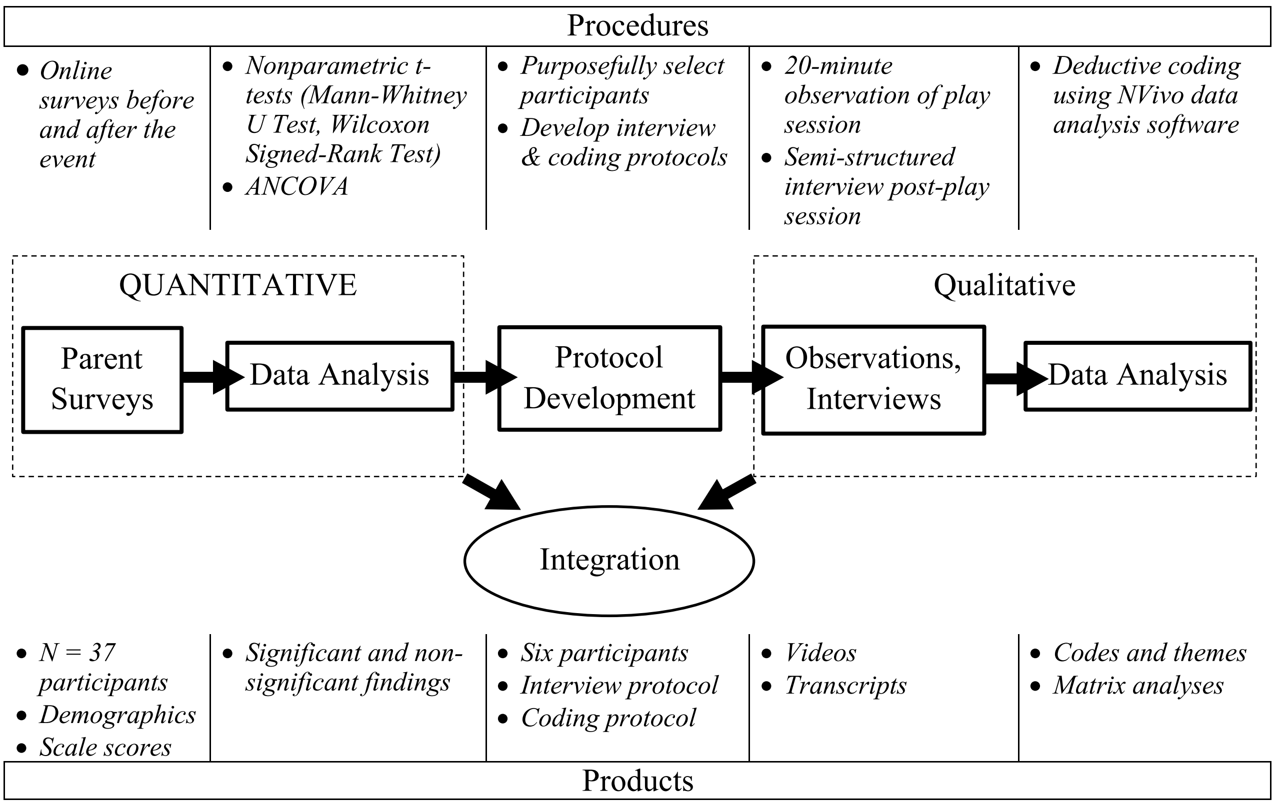
The findings on how families jointly engage with these platforms have important implications for the types of informal learning experiences parents will seek for their young children. Furthermore, understanding the impact of family-oriented programming on parent-child interactions in early childhood will contribute to the existing literature on joint media engagement and human-computer interaction research.

Chapter 4: Methodology

This thesis incorporates data collected from multiple studies: ScratchJr Family Days ($N = 58$ families), KIBO Family Days ($N = 51$ families), as well as a follow-up qualitative study involving both ScratchJr and KIBO ($N = 6$ dyads). Using the Family Coding Days dataset (i.e., the consolidation of data from both ScratchJr Family Days and KIBO Family Days), I performed secondary data analysis to explore differences in parents' experiences during ScratchJr versus KIBO Family Days. However, my aforementioned research questions could not be answered from the Family Coding Days study alone. The surveys collected during ScratchJr and KIBO Family Days captured parents' self-reported experiences quantitatively and at a broad level, which I sought to unpack further using qualitative analytic techniques. The use of both quantitative and qualitative methods, or mixed-methods research, provided a more holistic picture of the constructs in question than either method on its own (Creswell, 2014). Because I used and analyzed quantitative data first, which helped to inform how I collected and analyzed my qualitative data, the methodological approach used in this work parallels a mixed-methods sequential explanatory study design, though on a smaller scale (Creswell, 2014; Ivankova, Creswell, & Stick, 2006).

Figure 5 illustrates the mixed-methods approach with the emphasis (capitalized) placed on quantitative analysis (Creswell, 2014). The qualitative trends that emerged from the videotaped observations and interviews provided rich data that served to unpack and illuminate the quantitative findings. In the sections below, I describe how I chose my quantitative sample for secondary analysis and delineate the steps taken to develop the follow-up qualitative study.

Figure 5. Study Design



Sample

Because Family Coding Days were advertised as informal programming events for the whole family, a substantial portion of families attended with multiple adults and/or multiple children. Although this finding is interesting in and of itself, I wanted to explore specifically dyadic interactions, or the interactions between one parent and one child aged 5-7, which was the overlapping recommended age range for ScratchJr and KIBO. Furthermore, I wanted a consistent sample to compare across the Family Coding Days study and the follow-up qualitative study. Thus, I decided that my unit of analysis was a single parent-child dyad, in which the child was between 5-7 years old at the time of participation. Having a consistent unit of analysis between the quantitative and qualitative study was important for integrating results from both phases of the mixed-methods design. Because the Family Coding Days survey did not specifically capture whether parent respondents were reporting on all children brought to the event or just one of their children, the participants who attended with multiple adults and/or multiple children were excluded from analysis. Thus, after filtering out these participants, I had a sample of 40 families: 27 ScratchJr and 13 KIBO. However, three respondents did not answer any of the questions related to parent and child role engagement; due to these missing data, these three cases were excluded from analysis. The final analytic sample from the Family Coding Days dataset was 37 families (25 ScratchJr and 12 KIBO). Although the sample sizes were unequal, I was able to account for the imbalance by using non-parametric tests to explore differences between ScratchJr and KIBO Family Day events (Field, 2009).

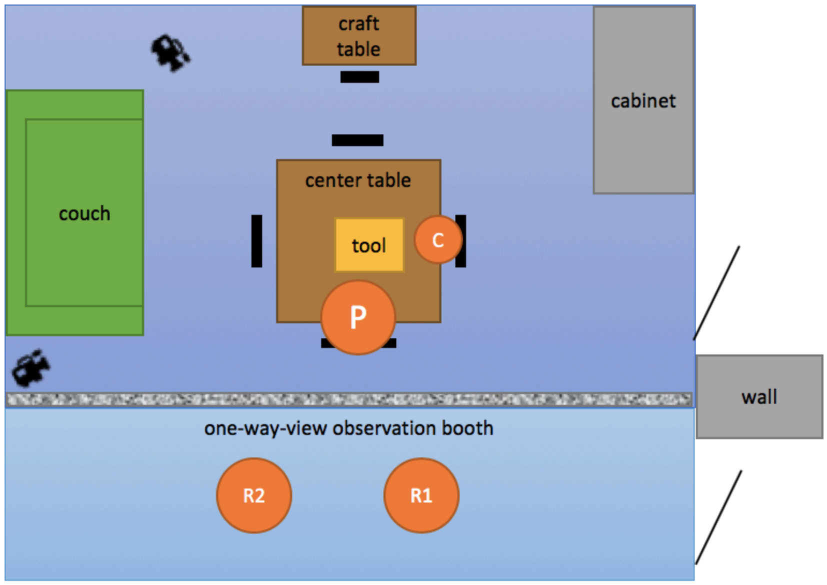
Study Flow

Once I had my final quantitative sample of $N = 37$ families, I proceeded to explore the data and perform secondary data analysis. I tested whether there were any demographic differences between the families who attended ScratchJr Family Days versus KIBO Family Days, as well as whether children and parents' reported role engagement differed for ScratchJr versus KIBO. Based on these quantitative findings (detailed in Chapter 6: Results), I developed a follow-up qualitative study for a more in-depth exploration of parent-child dyadic interactions using ScratchJr or KIBO.

For this qualitative study, I recruited six parent-child dyads for a one-hour family coding session, during which dyads were randomly assigned to either ScratchJr or KIBO and participated in a 20-minute videotaped play session with the tool. Families were recruited via emails to prior Family Coding Day attendees and DevTech summer program e-lists, as well as through the Eliot-Pearson Children's School. Parents completed brief surveys before and after the play session (using items from the Family Coding Days study) and participated in a semi-structured interview after the play session to reflect on their programming experience. Dyads were purposefully recruited with the following criteria: child must be between five and seven years old, and the parent must be able to complete surveys and converse in English for the duration of the study. Based on the quantitative findings, children's prior experience with ScratchJr or KIBO was also taken into consideration during participant recruitment.

The Parent-Child follow-up study protocol was developed using procedures similar to the Family Coding Days study but in a more experimental setting. The parent-child dyad entered a testing room (see layout in Figure 6), which was connected to an observation booth with a one-way-view mirror into the testing room. The observation booth had a built-in audio and video system, and an additional tripod was set up in the testing room; data were collected on both cameras in order to get multiple vantage points.

Figure 6. Parent-Child Interaction Study Room Layout



The parent first completed a pre-survey while the researcher allowed the child to freely explore the tool (ScratchJr or KIBO). Once the parent was done with the pre-survey, the dyad was given instructions about the informal 20-minute play session. The dyad was given two sample prompts from the Family Coding Days protocol (animal or play) or could come up their own idea for their coding project. One or sometimes two researchers observed the play session from the hidden observation booth and recorded live field notes. If the parent or child needed help at any point, they were to step outside the testing room, and the first researcher would come and assist them. After the 20-minute timer went off, the researcher came back into the room and conducted a semi-structured interview with the dyad to learn more about their joint programming experience. The interview protocol was developed using the Family Coding Days quantitative findings, specifically focusing on parents and children's behaviors and actions during the coding play session that constituted the five types of roles. For example, asking questions such as “Who came up with the project idea?” and “How did you choose the blocks for your program?” served to unpack the Planner role. A semi-structured approach allowed the researcher to probe and ask specific questions that would elicit a greater understanding of parent-child interactions and role engagement. For example, the researcher asked broad questions such as “Tell me about your project. What was your favorite/the hardest part about working together?” and more specific questions about the tasks or ideas that each person contributed to the programming activity. Parents then completed a brief post-survey, which included the same five role questions for both parents and children on a 1-5 Likert-type scale. The full detailed protocol is detailed in Appendix A and was approved by the Tufts University Institutional Review Board through a modification of the original protocol #1612026. Data from the Family Coding Days study and the Parent-Child follow-up study were integrated during the analysis phase, which is described in this next section.

Chapter 5: Analysis

Quantitative Data Analysis

Quantitative data from the Family Coding Days study were analyzed using the IMB SPSS Statistics Version 25 software. Pre-analysis screening indicated that three participants from the sample of 40 single parent-child dyads had missing scores on almost all role variables. Thus, these cases were excluded from analysis, resulting in a sample size of $N = 37$ dyads (25 ScratchJr and 12 KIBO). Frequencies and descriptives were examined for all categorical variables (i.e., family demographics) and continuous variables (i.e., role variables). I used Pearson’s correlation to test the bivariate relationships among all role variables to ensure that roles were not too highly correlated, which may indicate that participants were not able to distinguish the roles and thus were responding similarly to those items. The assumption of normality was not met for all ten role variables (Shapiro-Wilks test was significant, $p < .05$). There was high variation in the Parent Planner, Parent Teacher, Child Observer, Child Teacher, and Child Coach roles. The Parent Observer, Parent Coach, Parent Playmate, Child Planner, and Child Playmate roles were negatively skewed, suggesting that perhaps parents and children assume both similar and varying roles. Considering the skewed data and small sample size, nonparametric tests were used to explore differences in role engagement.

To explore the research question, “What roles are exhibited when children and parents jointly program using ScratchJr versus KIBO?”, I first assessed whether there were any demographic differences between the families who attended ScratchJr Family Days and families who attended KIBO Family Days using Chi-square tests. If there were significant demographic differences, those variables were used as covariates in subsequent analyses. I then explored whether ScratchJr families and KIBO families significantly differed in their role engagement using Mann-Whitney U tests (or ANCOVA for the purpose of including covariates). My final analysis explored whether parent roles significantly differed from child roles using Wilcoxon signed rank tests. These quantitative findings were unpacked using qualitative data from the Parent-Child follow-up study.

Qualitative Data Analysis

Qualitative data from the videotaped observations and the semi-structured interviews were transcribed and coded manually and then using the NVivo 12 qualitative data analysis software. The researchers recorded live field notes during the 20-minute play sessions, which were used to guide researchers in coding the transcripts. Transcripts were deductively coded using the five role categories for all 12 transcripts (six interviews and six play sessions): Planner, Observer, Teacher, Coach, and Playmate. Two DevTech student researchers assisted in coding the transcripts using the codebook in Appendix B. If all three researchers marked an excerpt or phrase from the transcript with the same role classification (e.g., Child Planner), the excerpt was coded in NVivo. If only one or two of the three researchers marked the same role classification for an excerpt, the team met to discuss and came to a consensus on whether the person did or did not engage in that role. Once all the codes were entered into the NVivo software, I aggregated the excerpts that were coded under the same role category in order to highlight qualitative examples of each child and parent role. For the significant roles that were identified using the Family Coding Days quantitative analyses, I aggregated those role examples and looked for similarities and differences in the way the roles were exhibited using ScratchJr versus KIBO. This comparison allowed me to explore my second research question, "What kinds of opportunities do ScratchJr and KIBO provide for family-oriented programming activities?". In particular, I looked for qualitative examples of how the design features of ScratchJr and KIBO enabled the parent-child dyads to engage in the programming activity together.

Chapter 6: Results

RQ1: What roles are exhibited when children and parents jointly program using ScratchJr versus KIBO?

Table 3 shows the demographic characteristics of the 37 families who attended ScratchJr or KIBO Family Days as single parent-child dyads. Chi-square tests indicated that child's prior experience was significantly associated with event type, $\chi^2(1) = 6.57$, $p = .01$. Only one of the 12 children had some experience with KIBO prior to attending a KIBO Family Day, whereas about half the children (52%) had experience with ScratchJr prior to attending a ScratchJr Family Day. There were no significant differences in other demographic characteristics (e.g., child age, parent gender, parent education, parent STEM/non-STEM background, and parent's prior tool experience) for families who attended ScratchJr versus KIBO Family Days.

Table 3. Demographics of Parent-Child Dyads at Family Coding Days

Demographics	**ScratchJr** (*n* = 25)	**KIBO** (*n* = 12)
Child Age		
5 years	7 (28%)	3 (25%)
6 years	9 (36%)	3 (25%)
7 years	9 (36%)	6 (50%)
Parent Gender		
Male	5 (20%)	2 (16.7%)
Female	20 (80%)	10 (83.3%)
Parent Education		
High school degree or equivalent	1 (4%)	0 (0%)
Some college, no degree	0 (0%)	1 (8.3%)
Associate degree	2 (8%)	0 (0%)
Bachelor's degree	8 (32%)	4 (33.3%)
Master's degree	10 (50%)	4 (33.3%)
Professional degree	4 (16%)	3 (25%)
Parent in STEM Profession		
Yes	10 (40%)	4 (33.3%)
No	15 (60%)	8 (66.7%)
Prior Experience with ScratchJr/KIBO		
Child has prior tool experience	13 (52%)	1 (8.3%)
Parent has prior tool experience	4 (16%)	1 (8.3%)

Table 4 displays the bivariate relations between all ten continuous variables: five parent roles and five child roles. All role variables were positively correlated with one another (*r* ranging from .029-.823). In order to account for the risk of Type I error due to multiple comparisons, the Bonferroni correction was applied, and the adjusted alpha value for determining statistical significance for the correlation matrix was .05 / [(10*9) / 2] = .0011. No two role variables were significantly correlated at the .001 level.

Table 4. Correlation Matrix of Parent and Child Roles

	P-P	P-O	P-T	P-C	P-L	C-P	C-O	C-T	C-C	C-L
Parent Planner (P-P)	1	.55	.74	.45	.22	.28	.63	.74	.73	.27
Parent Observer (P-O)	.55	1	.48	.43	.25	.33	.43	.40	.43	.13
Parent Teacher (P-T)	.74	.48	1	.48	.15	.24	.59	.49	.57	.05
Parent Coach (P-C)	.45	.43	.48	1	.44	.56	.38	.41	.44	.21
Parent Playmate (P-L)	.22	.25	.15	.44	1	.06	.26	.28	.36	.72
Child Planner (C-P)	.28	.33	.24	.56	.06	1	.34	.43	.45	.03
Child Observer (C-O)	.63	.43	.59	.38	.26	.34	1	.63	.61	.25
Child Teacher (C-T)	.74	.40	.49	.41	.28	.43	.63	1	.82	.35
Child Coach (C-C)	.73	.43	.57	.44	.36	.45	.61	.82	1	.40
Child Playmate (C-L)	.27	.13	.05	.21	.72	.03	.25	.35	.40	1

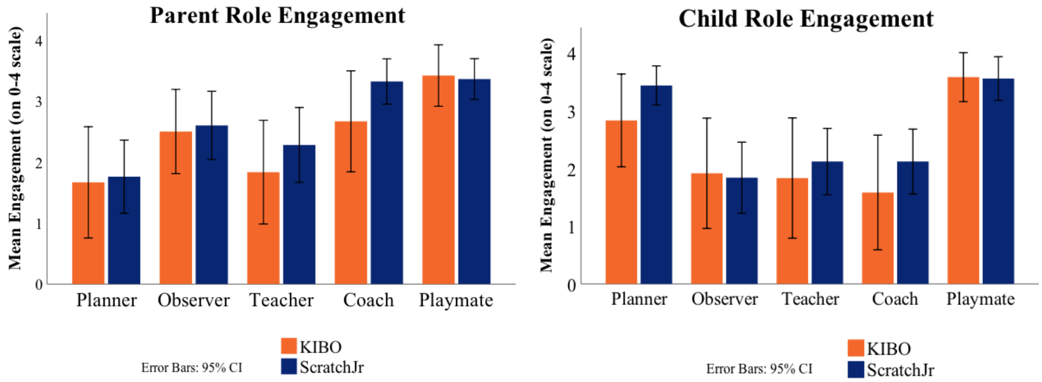
Table 5 displays the mean engagement of parents and children in the five role categories (Planner, Observer, Teacher, Coach, and Playmate) split by ScratchJr and KIBO. Because the data were skewed and the two sample sizes were small and unequal, Mann-Whitney U tests were used to test whether ScratchJr and KIBO families differed in their parent and child role engagement. Due to running multiple comparison tests which increases the risk of Type I error, I applied the Bonferroni correction, and the resulting alpha value was determined to be .05 / 10 = .005. Mann-Whitney U tests revealed no significant differences between ScratchJr and KIBO families' reported role engagement, $p > .05$ (see Table 5). Figure 7 provides a visual representation of this non-significant finding.

A one-way ANCOVA was conducted to determine the effect of interface on families' role engagement, controlling for children's prior experience with the interface. Even after controlling for child's prior experience, there was no significant effect of interface on families' reported role engagement, $p > .05$ (see Table 5).

Table 5. Mean (SD) Role Engagement by Interface

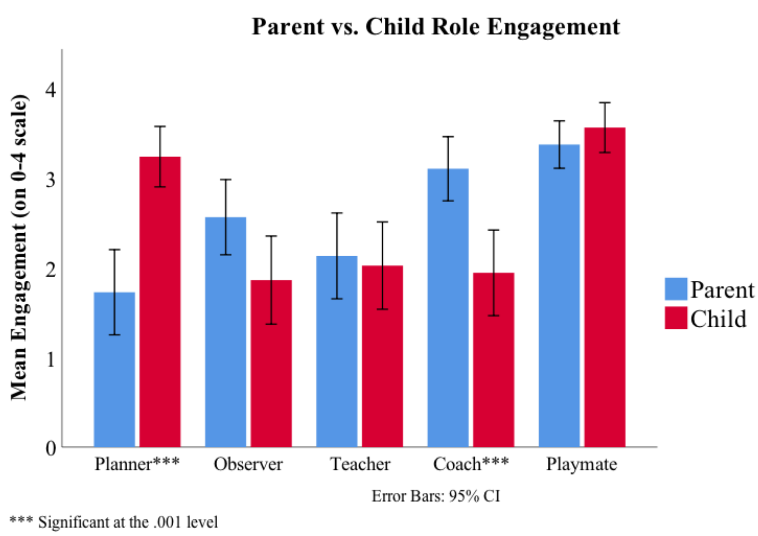
Role	Participant	ScratchJr Family Days (n = 25)	KIBO Family Days (n = 12)	Mann-Whitney		ANCOVA	
				U	p	F	p
Planner	Parent	1.76 (1.45)	1.67 (1.44)	155.0	.89	0.05	.83
	Child	3.44 (.82)	2.83 (1.27)	191.5	.18	1.44	.24
Observer	Parent	2.60 (1.35)	2.50 (1.09)	165.0	.64	0.23	.63
	Child	1.84 (1.49)	1.92 (1.51)	144.5	.86	0.14	.71
Teacher	Parent	2.28 (1.49)	1.83 (1.34)	176.5	.40	2.02	.17
	Child	2.12 (1.39)	1.83 (1.64)	163.0	.69	0.00	.99
Coach	Parent	3.32 (.90)	2.67 (1.30)	194.5	.15	1.64	.21
	Child	2.12 (1.36)	1.58 (1.56)	182.5	.30	0.19	.66
Playmate	Parent	3.36 (.81)	3.42 (.79)	145.0	.89	0.31	.58
	Child	3.56 (.92)	3.58 (.67)	157.0	.84	0.22	.64

Figure 7. Parent and Child Role Engagement by Interface



Wilcoxon signed-rank tests were used to test whether parents' self-reported role engagement differed from children's reported role engagement, regardless of interface. Again, due to running five comparison tests, the Bonferroni correction was applied, and the alpha value was determined to be .05 / 5 = .01. As indicated by Figure 8, children engaged highly as Planners compared to their parents, $Z = 444.00$, $p < .001$, and parents engaged highly as Coaches compared to their children, $Z = 5.00$, $p < .001$.

Figure 8. Parent versus Child Role Engagement



The non-significant finding that families' role engagement differed by type of interface and the significant finding that parents and children exhibited differing roles were further explored through the Parent-Child follow-up study. Six dyads were recruited and randomly assigned to the KIBO and ScratchJr conditions (see Table 6). Three of the six children had prior experience with the tool, whereas only one parent had prior experience with the tool. All but one parent did not work in a STEM-related profession. Detailed summaries of the six dyads' coding play sessions are included in Appendix C. The findings in this next section provide deeper insight into parent-child dyadic interactions with KIBO and ScratchJr and serve to unpack the quantitative findings.

Table 6. Parent-Child Follow-Up Study Participants

		Child			Mother			
	Condition	**Age**	**Gender**	**Prior Tool Experience**	**Age**	**STEM Profession**	**Education**	**Programming Experience**
Dyad 1	KIBO	6	M	No	40	Yes	Master's	Statistical software experience
Dyad 2	KIBO	6	F	Yes	42	No	Master's	None
Dyad 3	ScratchJr	6	M	No	41	No	Bachelor's	Some ScratchJr experience
Dyad 4	ScratchJr	7	F	No (but has extensive KIBO experience)	37	No	Bachelor's	Some KIBO experience
Dyad 5	KIBO	5	M	Yes	50	No	Master's	None
Dyad 6	ScratchJr	6	M	Yes	37	No	Master's	None

RQ2: What kinds of opportunities do ScratchJr and KIBO provide for promoting these roles?

The Family Coding Days analyses indicated that parents and children assumed multiple roles during the course of these events, and that role engagement did not significantly differ when using ScratchJr versus KIBO. Tables 7 and 8 document qualitative examples of parents and children engaging in each of the five roles: Planner, Observer, Teacher, Coach, and Playmate. Firstly, it is important to note the spatial arrangement among the child, parent, and tool in the study room. Two ScratchJr dyads sat on the couch with the parent holding the tablet, and one ScratchJr dyad sat at the center table with the tablet held between the dyad but mostly facing the child. This latter child (C6) was also the only child in the ScratchJr condition who had previously attended a week-long ScratchJr camp and had extensive prior experience with the app. The three KIBO dyads sat at the center table, but two dyads used the floor to test out their final KIBO programs. All the blocks were placed on the center table with equal access by the child and parent, but for the most part, the blocks faced the child.

Table 7. Qualitative Examples from KIBO Play Session and Semi-Structured Interview

Role	Dyad 1	Dyad 2	Dyad 5
Planner	C1: "I wanna make it like an animal… I'm gonna make it fake… fake animal like it sounds like 'ROARRRRR!'" P1: "This is the story of driving in Boston traffic. You're driving forward and you see a red light, and you're frustrated, and you scream. Beep beep! (Laughs) And then a little music playing. Sigh. 'Relax, relate, release.'" C1: "Then the hotel and the car… it actually makes sense if I add them together cuz people are trying, cuz one person is trying to get to the hotel and it shut down early so she wants to go really fast and then she doesn't miss the hotel."	C2: "Can we make a rainbow?" P2: "Yeah" C2: "I'm first gonna get red… You make a momma rainbow and I make a baby." P2: "What should I make? How about a smile? … Oh what if we put it in here?" C2: "No wait wait wait!!! No!" P2: "No? Do you want it to be pink? A star?" C2: "I wanna put this four times… On the next one, I'll show you what I'm gonna do."	C5: "Uh I wanna do… an animal" P5: "Animal? What animal would you like to do?" C5: "Tiger" C5: "I definitely want to go forward… I want backwards… I'm making a new sound." P5: "He actually double scanned on purpose because he wanted this to go more than once."
Observer	P1: "I was watching you [scan]."	R: "Who got to scan the program?" P2: "Well of course, she did! (Laughs) I just had to watch that part."	C5: "[I'm gonna] take this piece of paper and…" (Starts sifting through the construction paper) P5: (Watching him) "Oh okay are you gonna…"
Teacher	P1: "Do you want me to explain what all the blocks say on them? That way, you can understand your options… Let's sound this out. What does it look like? C1: "Sss… ppp… in. Spin." P1: "Spin! So you know what I think? I think the blue blocks are all about movement." P1: "I always like when he can teach and share things with me."	P2: "You have to tell me what to do." C2: "You add this to this block." P2: "So keep all this the same but add more on?" C2: "Yeah, but you have to always keep this and this. (Points to BEGIN and END blocks) The green one has to go always at the front, and the red one is at the end." P2: "Okay watch. You can't put your arm in front of the scissors because you'll get hurt."	P5: "Oh boy, I don't know what a tiger looks like. Does it have whiskers?" C5: "Yeah" P5: "Does it have freckles?" C5: "No, but it has stripes on its face" C5: "The tape is so hard." P5: "Yeah Mommy likes to do it at an angle [Demonstrates how to tear the tape] like that. I'll show you again."
Coach	P1: "Maybe you should come a little closer… maybe come from a different angle, maybe a little higher?" C1: "[She] keeps me like easy and balanced. Like saying don't give up and trying new things."	C2: "It's a blue light." P2: "Blue light (Laughs) cool, good job!" P2: "Watch your fingers, watch your fingers!"	P5: "Oh perfect! Good idea! … Nice straight line there… Nice trying. If at first it doesn't work, try try again."
Playmate	C1: "The end. We did it, Mom!" P1: "We did it! How do you like it?" C1: "It's pretty good." C1: "Hey Mom, do you want to try programming?" C1: "Oh Mom, let's see what your program does…What kind of block do you want? … What do you think you need first?"	P2: "You need help? … You wanna cut out the rainbow, and I'll help you do that part?" C2: "Scan" (Points flashing barcode light at herself) P2: (Laughs) "Oh it's scanning you?" P2: "We both started making separate rainbows…and then we combined them together."	P5: "Since you decorated it, how about I decorate it? Yeah?" P5: "Okay ready? I'm excited! I'm excited to see what we created!" C5: "It's flying" P5: "It's a flying tiger! High five, that was awesome."

Table 8. Qualitative Examples from ScratchJr Play Session and Semi-Structured Interview

Role	Dyad 3	Dyad 4	Dyad 6
Planner	C3: "No I wanna pick a new guy first." P3: "You can just add, make him [the wizard] be invisible and then visible and then invisible again."	P3: (After suggesting 3-4 other characters) "So what will be do then?" C3: "The… the silly chicken!" P3: "Let's do the silly chicken."	C6: "How about 'once upon a time, there was a wizard' … no actually you know what, let's do the scenery first… oh you know how we're doing a play? How about a theatre?"
Observer	P3: "Can you show me how to do actions because I didn't really catch that. I haven't really done this before."	P4: "I mean, I would like press all the buttons if it was just me for sure, but it's easy to like hold it and wait for her to do it." P4: (Watches C4 drawing cat on construction paper)	P6: "What are you trying to do?" C6: "Delete"
Teacher	C3: "If you press that, you can see how many times she should walk. Press 9 then 9 there and then she goes. And then she goes 9 steps" C3: "You take it like that and then swipe it back up and then you can hear that sound and that's done." C3: "How do you make it your own face?" P3: "Oh well you pick a body so which body do you like?"	P4: "Yeah press down right on top of the cat until the X appears and then click it. Goodbye cat" P4: "What does this orange button do?" C4: "It repeats!" C4: "One of the cat's ears is tilted another way so it can hear something play" P4: "That's why cats' ears are long like that"	P6: "You know how to spell tree." C6: "T… R… double E's. Tree beard." P6: "space… B… E… A.. rrrddd." C6: "R, D?" P6: "Nice!" C6: "We can do dragon avengers!" P6: "There's dragons in there?" C6: "I'll show you"
Coach	P3: "Oh there you go! Knew you'd figure it out." C3: "Should we make it repeat all over and over again, or should we end it?" P3: "What do you think?" C3: "Repeat all over and over. Let's see what that does." P3: "Why don't you come up with something he should say? Your ideas are so original."	C4: (Spontaneously sings a song about the characters) "When the silly chicken met the tulip, they said hi! But that was not all…" P4: "Would you like to record the song to go with your…" C4: "Yeah." C4: "I think I can keep the red mustache." P4: "I love it!"	C6: (Records sound but messes up in the middle) "I'm so embarrassed now!" P6: Maybe we shouldn't have said 'hold on a second'. Maybe we should say something else, or we should practice what we're gonna say." P6: "Nicely done!" P6: "I think you had the ideas and I just gave you other things to help with that idea right?"
Playmate	C3: "Momma… now you get to pick." P3: "How about we do it 22 steps? Let's see what happens." C3: "He said, 'ha ha!' I love when I made him say that." P3: "Yeah that was funny." R: "So who came up with this idea?" C3: "We both did." P3: "… You were really excited about having an invisible wizard and I came up with the dance for the fairy…"	P4: "Is it a nighttime chicken or a schoolhouse chicken? Is it a savannah chicken or a SPACE chicken? (C4 laughs) maybe it is! Space chicken! Check!" C4: "My cat is happy, see?" P4: "I bet she'll be really happy if she catches that chicken on the moon." C4: "Bet she'll not be happy with that chicken on the moon because I have a tulip as a friend." P4: "Oh! You think the tulip will defend the chicken against the cat?"	P6: "Ooh that's weird. I don't know if I like that, but if you like it, that's cool." C6: "I like it." P6: "Alright, should we keep the tail that color too?" P6: "We gotta get rid of the wings" C6: "And the tutu. Also those shoes! Legolas does not wear those!" P6: "Well we can pretend he does…Let's just pretend that's where his bow and arrow are." C6: "Okay."

The three dyads interacting with KIBO assumed multiple roles during the course of the play session, but there were qualitative differences in their dyadic interactions. For instance, halfway through the first dyad's play session after the child (C1) had finished creating his KIBO hotel decorations and corresponding program, C1 encouraged his mom (P1) to create her very own KIBO program, remarking, "Let's see how it goes, Mom!" As P1 purposefully chose blocks for her KIBO program, C1 remained fully engaged, even testing her ("What do you think you need first?") and providing words of encouragement as she scanned the blocks with his assistance ("Perfect"). The parent exemplified the role of Playmate in that she had all the same experiences as the child himself during the play session: coming up with a project idea, choosing blocks for the program, scanning the blocks, and testing the program. Contrast this parent-child interaction, in which both C1 and P1 were new to KIBO, with that of Dyad 5, in which only the child had extensive prior experience with KIBO. At first, C2 took time to teach her mom what the blocks meant and how they should be scanned; when they began to work on their decorations, C2 had exclusive control over how the decorations would be attached to the art platform. There were seven different instances where C2 firmly refused to allow P2 to touch the KIBO ("No wait!") or accept her mom's help. P2 exemplifies the roles of Observer ("I'll wait until you're done") or a more passive Playmate whose primary job was to be her child's helping hand.

There were also similarities and differences in the way the three ScratchJr dyads interacted with the graphical platform. Because the ScratchJr app has a set of existing characters, the first step for all three dyads was to scroll through the characters to decide what their coding project would be about. Although all three parents seemed to let children "drive" the project by choosing the characters, there was some variation in how parents tried to limit their child's time spent modifying the appearance of characters, instead wanting to focus on the programming aspect. Whereas P3 and P4 encouraged their children to begin programming their first character within five minutes into their coding play session, Dyad 6 did not start programming their four "Lord of the Rings" characters until the final few minutes. However, because C6 had extensive prior experience with ScratchJr, he knew exactly which blocks he wanted to use for his program and was able to execute his ideas fairly quickly. Because ScratchJr is a single-touch interface, meaning that only one finger can be used to manipulate icons on the screen at any given time, the three ScratchJr dyads often took turns directly using the interface, allowing both the child and parent to work collaboratively and actively as Playmates. For instance, when C4 started spontaneously singing a song about their chicken and tulip characters, P4 helped her record the song using the "Record Sound" block. Another example is when C6 scrambled to switch out all the "end" blocks for "repeat forever", P6 offered to separate the "end" block from the rest of the code and swipe the icon away to remove the block, allowing C6 to then drag down and connect the correct "repeat forever" block.

The Family Coding Days analyses indicated that regardless of interface, children engaged highly as Planners, whereas parents engaged highly as Coaches. These roles were unpacked in the six dyads in the Parent-Child follow-up. All the Planner codes for the three dyads who engaged with ScratchJr and the three dyads who engaged with KIBO were aggregated and classified into three sub-codes: *planning the project, planning the program,* and *planning the artistic elements*. Table 9 displays how these three planning categories emerged with ScratchJr versus with KIBO. It was interesting to note that all three ScratchJr dyads began first with selecting at least one character and the background and then moved onto programming the characters (though at different time points). All dyads had more than one character in their program. Children assumed the role of Planner by determining which and how many characters would be used, although parents offered suggestions or asked the child whether they wanted to add a new character to their project. Children also made decisions about which blocks to use for their characters' codes. For example, C3 asked his mom, "Should we make it repeat all over and over again, or should we end it?" to which she responded, "What do you think?". C3 is quick to reply, "repeat all over and over."

Unlike ScratchJr where children engaged in planning the artistic elements first, the three KIBO dyads differed in their planning process. Dyads 1 and 2 started with the programming aspect and then moved onto the decorating, whereas Dyad 5 began with decorating the art platform and then moved to programming. In addition, there were qualitative differences in the way children planned their KIBO programs. For instance, C2 assembled the blocks purely based on whether the pegs on the blocks fit seamlessly into the holes, even removing the "Shake" block from the program because it did not fit snugly with the others, stating simply, "it doesn't work". Conversely, C5 was particular about placing the "Light" blocks in the middle of the program before the "Play Sound" blocks to resemble the tiger blinking its eyes before it growled. Both methods of assembling the program were intentional and thus exhibited goal-oriented programming, but the goals were different. C2's goal was to have a perfectly linear program without any awkward gaps between the blocks, whereas C5's goal was to create a program that would align with the animal decoration co-created by him and his mom.

Table 9. Characteristics of the Child Planner Role

Sub-Codes	Child Planner: ScratchJr	Child Planner: KIBO
Planning the project	• Choosing an existing character or creating their own character • Choosing the total number of characters in the project	• Choosing a project topic by picking an animal or object
Planning the program	• Physically dragging blocks into the programming area or removing blocks by swiping up • Changing the number of steps on a blue motion block • Choosing between blocks (END versus REPEAT FOREVER, POP versus RECORD SOUND)	• Scanning a block multiple times on purpose • Assembling blocks that physically fit together seamlessly • Choosing blocks that resembled the animal's actions (lights = blinking eyes) • Choosing between parameter cards for the repeat loop (REPEAT 3 TIMES versus REPEAT FOREVER)
Planning the artistic elements	• Modifying the colors of the characters • Adding a background	• Choosing the color of construction paper • Deciding which platform piece to use and how to attach it to KIBO

The Parent Coach codes were similarly aggregated and unpacked, revealing four sub-codes: *praising, supporting child's autonomy, offering suggestions,* and *regaining child's interest* (see Table 10). Parents' socioemotional scaffolding behaviors were similarly present in ScratchJr and

KIBO dyads. For instance, parents in both conditions would verbalize their child's ability to think creatively ("Your ideas are so original") and to solve problems ("I knew you'd figure it out"). Parents would also often praise or encourage their child's effort taken to complete a task ("Nicely done" or "Good job"). This type of praise was particularly evident when a task was completed solely by the child, such as when the child tested out a completed program or put finishing touches on a character's appearance.

This tendency to support children's autonomy was a second prevalent feature of the Parent Coach role. Parents in both conditions permitted their children to make final decisions about what their coding project would be about. If children were stuck or indecisive, parents would assist by reading aloud the character/block names. For instance, when C4 was looking for a suitable background for their "silly chicken" ScratchJr project, P4 read aloud the different background options: "Is it a nighttime chicken or a schoolhouse chicken? Is it a savannah chicken or a space chicken?" Likewise, when C1 was deciding which blocks to use for their KIBO program, P1 offered to read aloud the names of the different blocks, asking, "Do you want me to explain what all the blocks say on them? That way, you can understand your options."

A third salient feature of the Parent Coach role was offering suggestions to help the child move the project along. For example, when C1 was struggling to scan the blocks using KIBO's embedded barcode scanner, P1 offered the suggestion to hold the KIBO robot from a different angle or to move it closer to the barcodes. Another example is when P5 offered the option of creating the tiger's tail by rolling up a piece of construction paper instead of cutting it out (although C5 did not end up using this suggestion). Parents in the ScratchJr condition similarly offered suggestions to improve upon their coding project. For instance, P6 pointed out the “undo” button, so that whenever the dyad drew or colored in the wrong area with “their fat fingers,” they were able to undo their actions easily. This type of editing was evident with the sound recorder, as well. Both KIBO and ScratchJr dyads re-recorded their sounds on the suggestion of their parents.

The fourth and final sub-category of the Parent Coach role was regaining the child’s interest in the middle of their coding project. During the 20-minute coding play session, there were several instances where children got distracted by random things in the room (e.g., the automated lights turning off, the mirror covering the one-way-view, the arts and crafts materials, etc.). Parents would redirect their child’s attention back to the coding project using a variety of tactics. For instance, parents would request their children to come look at the blocks or the completed program together, using the strategy of *joint attention*. In another case, C4 got distracted by the arts and crafts table in the middle of working on their ScratchJr project. To regain her child’s interest in the coding app, P4 playfully pretended as though she was “messing it up”. When C4 successfully returned and they played their program one last time, P4 noticed her child “feels inspired to draw” and encouraged her to make a drawing to go along with their chicken.

Table 10. Characteristics of the Parent Coach Role

Sub-Codes	Parent Coach	
	ScratchJr	**KIBO**
Praising	• Verbalizing child's ability to think creatively and solve problems • Praising child's effort	• Verbalizing child's ability to think creatively and solve problems • Praising child's effort
Supporting child's autonomy	• Encouraging child to choose characters/blocks • Holding tablet so that child can easily navigate the screen	• Encouraging child to choose project topic/blocks • Reading aloud the block names so that child can self-assemble program
Offering suggestions	• Pressing the "undo" button if someone accidentally made a mistake • Re-record sound • Asking whether to add a new character/block	• Scanning the KIBO from a different angle or moving the KIBO closer/farther • Providing ideas for which blocks to use
Regaining child's interest	• Pretending to mess up their project • Requesting to look at the final program together	• Requesting to look at the blocks together

Chapter 7: Discussion

The purpose of this thesis was to explore family-oriented programming in early childhood using two different programming technologies: the graphical ScratchJr app and the tangible KIBO robotics kit. Specifically, this thesis sought to identify the roles exhibited by children and parents when co-engaging with these platforms, as well as to explore the opportunities these platforms provide for promoting these roles. Parents who participated in the community-oriented Family Coding Day events and the Parent-Child coding play sessions reported engaging in multiple roles while co-engaging in ScratchJr or KIBO programming activities with their children. This finding supports previous research on Family Creative Learning workshops, which also indicated that families assume and develop different roles over the course of these events (Roque et al., 2014, 2016). Although Family Creative Learning workshops primarily served families with children in late elementary and middle school, the findings from this work show that family-oriented programming in early childhood offer unique opportunities for young children and parents to interact and learn from one another.

The first hypothesis that families would assume different roles when jointly engaging with ScratchJr versus KIBO was not supported by the quantitative findings from the Family Coding Days study. Although more children had previous tool experience prior to attending a ScratchJr Family Day as compared to a KIBO Family Day, this variable did not impact children or their parents' reported role engagement. This finding contradicts previous research studies that show that families' interactions tend to be more child-directed when the child has previously used the tool before (Roque et al., 2014, 2016). In addition, parents' STEM or non-STEM background did not impact whether they attended a KIBO or ScratchJr Family Day, or the extent to which they engaged in each of the five roles. This finding contradicts previous findings that indicate that parents' background in an IT or STEM-related profession impacts the way they co-engage in computing activities with their children (Barron et al., 2009; Bers et al., 2004; Lin & Liu, 2012; Roque et al., 2014). These contradictory findings suggest that regardless of children or parents' prior coding background or experience doing similar activities, the activity of creating a coding project with ScratchJr or KIBO remains open-ended and accessible enough for all kinds of learners. Furthermore, these findings also suggest that in early childhood, perhaps prior coding experience from school or work may not be as important as how children and parents typically interact in informal settings. For instance, one parent commenting that "[my child] and I, we do everything together, like we drive together and grocery shop together" demonstrates how young children coding together with their parents may be related to how they engage in other activities together. Future work should explore these connections.

The second hypothesis that children would assume different roles than parents during family-oriented programming activities was supported by these findings. The quantitative phase revealed that children engaged highly as Planners as compared to parents, whereas parents engaged highly as Coaches as compared to their children. These differences suggest that child and parent roles during family-oriented programming parallel the drivers of joint media engagement, specifically the child's desire to engage with the tool and the parent's desire to engage the child (Joan Ganz Cooney Center, 2014).

When the Child Planner role was unpacked further in the follow-up qualitative phase, there were some qualitative differences in the way children planned their coding projects with each type of interface. With ScratchJr, children's planning of the artistic elements and the project topic came first because they needed to first choose the characters that they were then going to program. Although the ScratchJr app comes with the option of creating own characters, only C6 opted to do this for one of his four characters. With KIBO, children could either start with planning out their decorations (as C5 chose to do by creating the tiger) or assembling the program first (as C2 chose to do by looking for blocks that fit snugly). KIBO and ScratchJr are both open-ended coding platforms, but this difference in children's planning processes highlights a key difference between the interfaces: ScratchJr comes with a pre-existing set of characters to choose from, whereas KIBO is completely open-ended. This difference parallels previous findings with ScratchJr and KIBO, which show that children using ScratchJr would spend a majority of their time on the paint editor as compared to children using KIBO, who tended to vary their interests between the blocks and the art platforms (Pugnali et al., 2017). Regardless of interface, however, children were ultimately responsible for choosing their project topic and the blocks they would use to program their characters, unless they specifically chose to allow their parent to have a turn (as C1 did, for example).

The qualitative study also brought to light the various ways in which parents engaged as Coaches: praising, supporting child's autonomy, offering suggestions, and redirecting child's interest. P6 succinctly summarizes her role during the post-coding play session interview: "I think you had the ideas. I just gave you other things to help with that idea." Parents' roles as Coaches were more similar between the ScratchJr and KIBO dyads, the only salient difference being the specific ways in which they offered suggestions to move the project forward. With KIBO, one example of offering suggestions was P1 providing advice on holding the KIBO from a different angle to expedite the scanning process. With ScratchJr, one example of offering suggestions was P6 showing her child how to use the "undo" button whenever they messed up their drawing with "their fat fingers". These examples support existing literature on how children can benefit from parental support by being able to create more complex programs and engage in goal-oriented programming (Hughes & Greenhough, 1995; Lin & Liu, 2012). Furthermore, by assuming this Coach role, parents were able to assist their children in successfully performing tasks that children may not have been able to do on their own, lending support for Vygotsky's concept of zone of proximal development and the mediating role of parents (Clark, 2011; Vygotsky, 1980). The other strategies exhibited in the Parent Coach role, such as words and gestures praising the child's effort, supporting their autonomy, and redirecting their attention, relate closely to the tactics identified in the literature on joint media engagement. Socioemotional, cognitive, and behavioral scaffolding strategies are all ways in which parents facilitate children's engagement in technology-mediated activities (Connell et al., 2015; Takeuchi & Stevens, 2011).

The third hypothesis that ScratchJr and KIBO provide different opportunities for family-oriented programming was not supported by the quantitative findings, but the qualitative data illuminated how KIBO could provide greater points of access. Although parents reported engaging collaboratively during Family Coding Days (Govind, Relkin & Bers, manuscript submitted for publication; Relkin, Govind & Bers, manuscript in preparation), the *rules* regarding sharing the interface and *dividing project tasks* became more explicit in the Parent-Child follow-up (Engestrom, 1999). With ScratchJr, dyads needed to position themselves so that both the child and parent had visual access to the screen, and they took turns directly engaging with the interface due to the single-touch feature of the graphical platform. With KIBO, dyads would move materials around to create space for all the blocks and the moving robot. Not only would they jointly attend to the activity at hand, KIBO dyads could also jointly *use* the interface. For instance, one child would be assembling the art platform while the parent would be actively touching and examining the tangible programming blocks. This example highlights another key difference between ScratchJr and KIBO for multiple users: ScratchJr as a single-touch graphical platform allows for joint attention, whereas KIBO as a tangible platform with various parts allows for both joint attention and joint usage. This difference supports the current literature in the human-computer interaction field on the affordances of graphical and tangible programming interfaces, which point to tangible interfaces allowing for greater collaboration and engagement for multiple users (Horn et al., 2009; Sapounidis & Demetriadis, 2013; Xie et al., 2008). One particular advantage of tangible interfaces, as identified by Horn and colleagues (2009), is the affordance of multiple entry and access points. KIBO offers multiple entry points, which means children and parents can be attracted to the wooden blocks, the KIBO robot and sensors, or even the arts and crafts materials. These multiple parts also allow for multiple access points, which means the tool can be simultaneously used by both children and parents. Conversely, graphical interfaces such as ScratchJr has a limited, two-dimensional display space, which limits children's ability to share the interface and complete tasks (Xie et al., 2008). However, two of the three parent-child dyads in the ScratchJr condition were able to combat this limitation by positioning themselves on the sofa with the tablet placed between them.

Findings by Horn and colleagues (2009) also suggest that children were not only more actively engaged with the tangible interface, but that this engagement positioned parents to take on a more supportive role rather than an instructional role. In this work, there were no significant differences in families' role engagement between ScratchJr and KIBO, but the significance of the Child Planner and Parent Coach roles seem to support this previous finding (Horn et al., 2009).

Limitations

There are several limitations of this work, including sample size, parent report, self-selection bias, facilitator resources, and different sampling methods for the two study phases.

Because the focus of the study was limited to single parent-child dyads where the child was between the ages of five and seven, the analytic sample was smaller and prompted the use of nonparametric statistical tests. Additional cases were excluded because parents did not complete one of the surveys or attended both ScratchJr and KIBO Family Day events. Although the analytic sample of 37 dyads was enough to explore differences in role engagement between ScratchJr and KIBO families, a larger sample size may have shown greater variation and thus contributed to greater generalizability. In addition, there were only six dyads explored in the follow-up qualitative study. This sample was more than sufficient to provide qualitative examples of how dyads engaged with ScratchJr and KIBO, especially because the dyads had some variation in children's prior tool experience and parent's prior programming experience. However, the findings may be limited to these specific case studies.

Another limitation of the Family Coding Days study is that child and parent survey data were reported by parents themselves. Self-reported survey responses are prone to some level of bias, particularly social desirability bias, in which parents may over-report socially desirable traits and under-report undesirable traits (Nederhof, 1985). Although the Parent-Child follow-up study brought to light some qualitative examples of the quantitative findings, future research should include independent observation to validate the parent report measure. This step would ensure that parents' self-reported data were a valid representation of the findings and coincide with researchers' observations and analyses.

In addition to response bias, another limitation of this work is self-selection bias, which limits generalizability. Families self-selected to attend ScratchJr and KIBO Family Day events and participate in pre- and post-surveys for research purposes. Both event facilitators and families were recruited directly from ScratchJr, KinderLab Robotics, DevTech, and Family Code Night e-lists and social media platforms, as well as indirectly through their outside facilitators. Although recruitment methods varied among events, the analytic sample for this study was comprised of highly educated parents from middle-to-high socioeconomic backgrounds. Future research should explore whether families that do not belong to these demographic characteristics report similar Family Coding Day experiences.

Another study limitation is facilitator resources. In order for outside facilitators to be able to host Family Day events in their respective communities, they required a large enough space for families to come together, as well as access to the technologies themselves (i.e., tablets and/or KIBO robotic kits). Although ScratchJr is a freely downloadable app, the KIBO robotic kits are more expensive, and these events would require multiple kits to be shared among families. These financial and logistical reasons may be why KIBO Family Day events were hosted solely by the DevTech research team. Furthermore, because the Family Day protocols included a comprehensive set of resources so that facilitators could adapt them to meet their individual needs, the protocols were lengthy, which also could have limited facilitators' interest in actually hosting the event after receiving the protocol. In fact, less than 3% of respondents to ScratchJr or KIBO Family Day recruitment emails actually ended up hosting an event.

Finally, a methodological limitation of this study was that the six dyads who participated in the follow-up qualitative study were not a subset of the larger sample who participated in a ScratchJr or KIBO Family Day event. The typical sampling method for a sequential explanatory mixed-methods study design involves following up with a subset of the original quantitative sample (Creswell, 2014). However, recruiting an unrelated sample of parent-child dyads sufficed for this study because the purpose was simply to provide qualitative examples of parent-child interactions with the two coding platforms. In addition, the Parent-Child follow-up study took place in an experimental setting that was very different from the informal, community-oriented, and organic space of Family Day events. It is likely that parents and children's behavior is impacted by contextual differences between these spaces. Studies show that both adults and children tend to act differently when they know they are being watched, or in this case, videotaped (Rideout, 2017). Nevertheless, the close examination of parent-child dyads jointly engaging with ScratchJr and KIBO provided rich qualitative data that was not captured from the larger community-oriented events.

Future Directions

Although this thesis has its limitations, it provides exploratory data for how children and families interact with graphical and tangible technologies. As increasingly more schools, states, and countries adopt K-12 computer science standards and frameworks, the question of how to engage families' learning at home and through other informal means will become more critical. In addition to the aforementioned possibilities for future work, research should explore how ScratchJr and KIBO are used in formal versus informal learning settings and the resulting impact on children's engagement, as well as learning outcomes. In this thesis, the Parent-Child follow-up study was conducted to highlight qualitative examples of the roles exhibited by children and parents. However, future research should look to explore how these roles facilitate children's learning outcomes, particularly their learning of computational thinking skills such as sequencing and debugging.

The self-selection bias discussed as a limitation of this study also introduces a new construct: motivation. What motivates parents to seek out programming opportunities and family-oriented experiences for their children? Do parents prefer children to engage with tangible or graphical interfaces, and might this preference change in the context of family-oriented programming? For instance, the American Academy of Pediatrics recommends limiting screen-time in early childhood, which may hinder parents' interest in promoting graphical interfaces. However, 64% of parents in a national survey reported co-using tablets with their children ages eight and younger (Connell et al., 2015). In this study, the findings suggest that both graphical and tangible interfaces offer positive experiences for families to engage in creative programming activities. Future research should look to include programming technologies in joint media engagement literature and explore the reasons why parents might promote or limit their children's exposure to these new types of playful learning tools.

Chapter 8: Conclusion

As K-12 computer science education becomes an increasingly important national and international priority in schools and other formal learning settings (Code.org, 2018), family engagement will also become increasingly salient. Family-oriented programming has emerged in recent years as a way to bring children and parents together to jointly engage in creative programming activities; however, most work has explored parent-child interactions and role engagement for children in late elementary years or older. There was a gap in understanding family-oriented programming in early childhood, a period of children's development where much of their learning occurs through play-based activities with caregivers (Bers, 2018; NAEYC & Fred Rogers Center, 2012; Rideout, 2017). This thesis served to fill this gap by identifying the roles exhibited by parents and young children during joint programming activities using different kinds of interfaces.

The literature on joint media engagement and human-computer interaction, coupled together in the visual framework of the activity theory model, served to highlight several factors that may influence parent-child dynamics during programming activities, such as spatial orientation, prior experience with coding, and background in a STEM-related field (Engestrom, 1999; Horn et al., 2009; Kaptelinin & Nardi, 2018; Sapounidis & Demetriadis, 2013; Takeuchi &

Stevens, 2011). Would parents and children interact differently based on the type of interface used? Were there specific design features of graphical and tangible interfaces that would foster greater collaboration and sharing behaviors? What roles would parents and children assume to divide tasks and successfully work together on their coding projects? Answers to these questions were explored in this thesis, which looked at how children ages 5-7 and their parents jointly engaged with ScratchJr (a graphical interface) versus KIBO robotics (a tangible interface), two developmentally appropriate, block-based coding platforms used widely around the world.

Using data from community-oriented Family Coding Day events with ScratchJr and KIBO, the findings suggest there were no quantitative differences in families' role engagement. Regardless of interface, children were reported to engage in planning roles, whereas parents reported engaging in coaching roles. Through a follow-up study of parent-child dyadic play sessions with ScratchJr and KIBO, the findings illuminated behaviors and actions that constituted the Child Planner and Parent Coach roles and identified several qualitative differences between dyads' usage of ScratchJr versus KIBO. Altogether, this mixed-methods study brings to light the roles exhibited in parent-child dyadic interactions with ScratchJr and KIBO. These findings have important implications for the types of new technologically-mediated experiences parents may seek for young children, which are discussed in this next section.

Implications for Practice

In the larger conversation of the graphical versus tangible debate and which might be considered more "playful" or suitable for multiple partners, this work suggests that both ScratchJr and KIBO foster positive experiences for young children and parents to jointly engage in creative computing activities. I offer the following reflections to support parents, educators, and practitioners seeking to promote family-oriented coding opportunities for young children.

Consuming versus creating with technology. The rise of new programming technologies that teach young children how to code provide new ways for children to think about the world around them (Bers, 2018). By programming stories on ScratchJr or creating robotic animals with KIBO, children not only interact with these tools but learn how to produce creative artifacts. Families should be encouraged to create projects that are meaningful and personal to them. Furthermore, just as the family literacy movement has shown how shared reading interventions and home reading programs may enhance children's linguistic and cognitive development (National Early Literacy Panel, 2008), family coding has the potential for similar impact on children's computational thinking skills. Parents can play an important role in facilitating children's creativity, personal expression, and problem-solving skills through the activity of programming together.

Need for adequate resources. How parents and children engage with different kinds of programming interfaces depends on the availability of resources, including time, environment, number of tablets or robotic kits, and facilitators. Family-oriented coding events should be long enough to allow children and parents to successfully complete a coding project from start to finish but concise enough to keep families fully engaged. Although projects can range in complexity and can always be improved and iterated upon, events should include opportunities for families to share feedback with one another. The event location and spatial arrangement of the tool are also critical factors that influence how families engage with the interfaces. In this study, we saw that families chose to sit on couches when using ScratchJr or on the floor to test out KIBO programs. Family coding events should ensure that the environment offers adequate and appropriate spaces for both children and parents to easily access the interface. In addition, having one tablet or KIBO kit per family might promote greater engagement than having to share the tools among multiple families, particularly with the graphical interface. The more users who have to share a single screen, the more difficult collaboration can become. Lastly, depending on the size of events, there may need to be multiple facilitators around to help families with any issues that may arise (e.g., changing KIBO batteries, helping families debug their program, etc.). Facilitators should be trained on how to use the interface and offer strategies for families. Just as parents took on coaching roles and allowed their children to drive and plan their projects, facilitators should also use scaffolding techniques to encourage families to use problem-solving skills to resolve any issues.

With this work and the freely available resources for hosting Family Coding Days with ScratchJr and KIBO (http://sites.tufts.edu/devtech/learn-with-us/for-children-and-families/), it is my hope that future work will continue to improve upon these models for engaging young children and parents in programming together.

Appendix A: Parent-Child Follow-Up Protocol

Introduction (5 min)

Prior to the visit, parent-child dyad will be randomly assigned to KIBO or ScratchJr. Research assistant, parent, and child all introduce themselves. Research assistant explains the purpose of the session: "Today, you and your parent are going to play with [KIBO/ScratchJr], which is a [robotics kit that is programmed with wooden blocks/a programming app where you can snap blocks together to animate characters on the screen]. Before you both get to play together, I'm going to first show you [KIBO/ScratchJr] while your parent fills out a short survey.

Parent Survey + Child Exploration of Toy (10-15 min)

Research assistant shows the educational technology to the child and allows child to explore. Research assistant gives minimal instructions unless being prompted by the child (e.g. "What does this block do?", "How do I change the color of my kitten?"). Parent, sitting next to the child, completes the pre-survey on a laptop. Research assistant takes notes of any interactions between child and parent during this time.

Explanation of Activity Prompt (5 min)

Research assistant: "Now that you've had a chance to explore the game a little bit, you and your parent are going to create a project together using [KIBO/ScratchJr]." Parent-child dyads will be provided both the Animal and Play prompts and will be invited to choose one for their project.

Animal prompt:

- **KIBO:** Lions, tigers, and bears, oh my! Create and program a robotic replica of your favorite animal using the KIBO robotics kit. Once you've programmed your animal's behaviors, don't forget to decorate your robot to look like the animal you've chosen using arts, crafts, and recycled materials.
- **ScratchJr:** Lions, tigers, and bears, oh my! Create and program an animal character on ScratchJr. Once you've programmed what your animal will do, don't forget to customize your background so that your animal has a habitat.

Play prompt:

- **KIBO:** Lights, camera, action! Program your KIBO robot to act out a scene from your favorite movie, book, or play. The order (or sequence) of the programming actions you choose will change the way your robot acts and moves. Don't forget to decorate your actor when you're all done!
- **ScratchJr:** Lights, camera, action! Create and program a character in ScratchJr to act out a scene from your favorite movie, book, or play. The order (or sequence) of the programming actions you choose will change the way your character acts and moves. Don't forget to customize the background and character when you're all done!

Research assistant: "While you're working on your project, there are going to be two video cameras recording here and here, but I want you to do your best to ignore them and pretend like they're not even there. I'm going to be right outside if you run into any problems or have any questions. After 20 minutes, I'll come back inside and see the project that you have created! If you finish before 20 minutes, you can add more things to your project or come up with a story for your project. If you don't finish by the time I come back, that's okay! You can share what you have created so far."

Parent-Child Play Session (20 min)

Research assistant starts videotaping and steps outside the room. Parent and child work together on the activity using the educational technology. Research assistant takes notes of any interruptions during the session.

Sharing the Project (15 min)

After 20 minutes, research assistant comes back into the room. Research assistant asks parent and child to share their project by asking the following questions. After the semi-structured interview, parent completes a brief post-survey on the laptop about their experience.

Semi-structured interview questions: These questions will serve as a starting point for the research assistant's conversation with the parent and child. The research assistant is encouraged to ask follow-up questions based on the parent and child's responses.

1. Tell me about your project. (Follow up: Who came up with that idea? How did you decide what your project was going to be?)
2. Who did which part of the project? (Follow up: How did you decide which blocks to use?)
3. What was the best part about working together? (Follow up: Why?)
4. What was the hardest part about working together? (Follow up: Did you come across any problems? How did you fix it?)

Appendix B: Parent-Child Role Engagement Codebook

Role	Definition	When to use	When not to use	Examples from pilot video
Planner	Planned out project topic and delegated tasks to members of the group	Takes notes or plans out project on a piece of paper Initiates a project topic Asks to include a new character or block to elaborate on the topic	Asking how a block or an art feature works – use teacher instead Asking questions to prompt the other person to choose a topic – use coach instead	**Parent:** We can either do something more with the background, or we can make him have a little animation, give a little voice or something. **Child:** I have a better idea… I'm looking for a person.
Observer	Let others guide project creation, did not contribute to the group's coding activities	Watches the other person interact with the interface without doing/saying anything for 5 seconds or longer	Talking or asking questions while the other person interacts with the interface – use playmate instead	**Parent:** Let me try this part. Can I do a little bit? I love coloring. (before this statement, parent was observing the child using the ScratchJr paint editor)
Teacher	Explained some of the coding topics to the group during the activity	Using words to describe the function of a block or use the art features Demonstrating how to attach blocks, use the paint editor, scan blocks, etc. Connecting to other curricular domains (e.g., spelling words, reading, counting)	Encouraging the other person to have a turn or supporting autonomy – use coach instead	**Parent:** Keep holding for it to go away. It gives you the X button, maybe right? **Parent:** You know how to spell Isla… I-S-L-A. **Parent:** Okay, now you hit the brown, you hit the circle. Do that move here, I think.
Coach	Encouraged and supported the group, offered suggestions to group members during the activity	Using words to praise or encourage effort Using gestures to praise or offer support (e.g., high five, pat on the back)	Laughing together or working together to troubleshoot– use playmate instead	**Parent:** I like how you're taking your time. **Parent:** That is amazing. I love it! **Parent:** High five! Nice job.
Playmate	Shared the fun, enjoyable parts of the activity	Asking to take turns or to do something together Laughing, making jokes	Using words/gestures to compliment the other person – use coach instead	**Parent:** Can you make a narwhal sound again because I kind of want to do the narwhal sound with you together.

Appendix C: Brief Case Portraits

Dyad 1: The play session begins with the child (C1) showcasing the hotel he had drawn on the whiteboard. He shows his mom (P1) how to scan the blocks, asking for her help in covering the other barcodes while he scans. They test the program and notice that KIBO does not perform all the actions in the program. C1 calls the researcher for help and after re-scanning and re-testing, the dyad realizes that they probably missed some of the blocks while scanning and that they forgot to record a sound on KIBO's Sound Recorder module. When the researcher leaves, P1 encourages C1 to think about what else they can do for their "KIBO hotel" project, and they work together to mount his construction paper hotel to the art platform. P1 continues to encourage C1 to explore KIBO, which prompts C1 to look through all the KIBO blocks as P1 helps him read aloud the block names. C1 assembles a new program with as many blocks as possible. Instead of scanning this new program himself, C1 asks P1 to try programming and scanning the program that he just assembled. C1 assists by covering the other barcodes with his hands. They move the blocks to the side before running the KIBO program so that there would be enough space. After the program runs successfully, C1 eagerly calls the researcher back to showcase their new program. The dyad asks for help on how to make a new KIBO project, this time maybe a fake animal. C1 takes some time to think about some animal sounds but ultimately decides to give P1 a turn in making her own KIBO project. P1 makes a KIBO car driving in Boston and purposefully chooses blocks to make her driving story come alive. C1 allows P1 to plan the project by herself but remains fully engaged as he helps her assemble and scan the blocks, even providing words of encouragement. P1 showcases her program, after which the dyad celebrates their accomplishment with a high-five.

C1's "KIBO Hotel" Decorations and P1's "Boston Driving" Program

Dyad 2: C2 already has a KIBO program assembled and shows her mom how to scan the blocks. C2 places the KIBO on the floor before running the program. Once she tests it out once, C2 offers P2 the opportunity to change the program, to which P2 asks her daughter for assistance, "you have to tell me what to do." C2 teaches her mom about starting and ending the KIBO program with the respective green and red blocks and excitedly holds up the "repeat forever" parameter card. P2 suggests some blocks while C2 takes over assembling the blocks and demonstrating how KIBO's actions change depending on which blocks are scanned. Within three minutes, C2 shifts her attention to decorating KIBO and asks P2 if they can make a rainbow. P2 agrees and again allows her daughter to take the lead. They switch out the platform piece, and C2 instructs her mom to assist her with cutting out construction paper. C2 originally decides that her mom will make a "momma rainbow" and that C2 will make a "baby". When cutting out the rainbow, C2 encourages her mom to cut on the inside of the circle, confidently reporting that her way "is the easy way." P2 offers her help with taping the rainbow, to which C2 refuses. After seeing her mom's big rainbow, C2 decides that she doesn't want to use her own and focuses her attention on taping the rainbow to the platform piece. Each time C2 steps away from their project to get tape or construction paper and P2 offers her assistance, C2 exclaims, "No wait!" C2 tells P2 to make something else for their project, like a star or heart, while C2 pokes holes in the rainbow so that the platform can be properly attached to the pegs of the motorized pedestal. Once P2 finishes cutting out a pink heart, she watches her daughter carefully as she pokes holes using the scissors, worrying about her safety. C2 insists that she does this job independently but accepts her mother's suggestions. When the dyad has about five minutes remaining, they had finished assembling the rainbow and heart decorations onto KIBO and return to programming. C2 focuses on assembling the blocks based on the physical ease of connecting them together. P2 remarks, "Oh it's just sticky. Want me to help you?" to which C2 does not respond and continues assembling. Their final program includes a repeat loop with the "4" parameter card, but C2 states that next time, she will change it to something different. During the interview, C2 adds more decorations to the KIBO and switches out the "4" parameter card to "forever".

Dyad 2's Final Project: "Rainbow and Heart"

Dyad 3: The play session begins with C3 choosing two characters for their project: a wizard and a seahorse. Before beginning to program the characters, C3 decides to switch the background to t a moon while P3 observes and comments on the number of characters and backgrounds to choose from. C3 adds a third character to their project: a fairy. P3 encourages him to start programming his characters and allows him to take the lead. The dyad talk through how they will first program the fairy to "do magic" and look through all the programming blocks, finally deciding on making the fairy move backwards and make a "whoop whoop" sound. C3 shows his mother how to change the number on the programming block so that the fairy moves backwards 22 steps. Since P3 recorded the "whoop whoop" sound, she offers C3 the opportunity to practice some sounds so that he could record one himself. Despite P3 encouraging him to try a sound, C3 seems more focused on adding new motion blocks and changing the number of steps, ending with a REPEAT FOREVER block. Once the dyad has a complete program, they test it out. P6 laughs and comments, "That's kinda weird." She asks if C3 wants to add a different character, but they decide that they will next program the wizard. The dyad works together to figure out which blocks they will use for the wizard's program; although C3 asks P3 her opinion on which blocks to use, the choice is ultimately his. P3's role seems to be guiding the project forward and prompting him to think about the larger story with the seahorse, fairy, and wizard characters. P3 assists C3 in creating the sequence of the wizard's program: "So why don't you have him be invisible and then have him become visible again and say, 'ha ha!' like he did a magic trick". As they continue to make changes to the wizard's program, P3 accidentally swipes up the program, which deletes their entire code and upsets C3. P3 assures him that she thinks she remembers what they had and helps him recreate the program. P3 takes more ownership of the tablet as they discuss which blocks they will keep the same as before and which ones they might modify. The dyad also modifies the number of steps on the fairy's program. Towards the end, C3 asks to include a new character, maybe one where he can include his own face. P3 attempts to help him find that character but gets confused, so she calls the researcher for help. C3 is eager to showcase their program to the researcher. The researcher shows the dyad the various characters they can use to insert their own face, and C3 ends up choosing the astronaut. C3 inserts a picture of his face inside the astronaut helmet and changes the color of the astronaut suit to pink. They delete the grandfather that they accidentally added to the program with help from the researcher. In the final minute or so, the dyad program the astronaut to move up and down and say "hi".

Dyad 3's Final Project: "The Wizard, the Seahorse, and the Astronaut"

Dyad 4: P4 reads aloud both the animal and play prompt for her child to decide. C4 chooses the play prompt but does not have an idea in mind. P4 offers some suggestions and encourages her to look through all the ScratchJr characters. C4 scrolls through and decides on a silly chicken. C4 edits the colors of the chicken with the playful encouragement of P4, and together they get rid of the ScratchJr kitten that was originally in their project. P4 suggests if they should give the chicken a background and starts reading off the names of the different backgrounds. The dyad gets excited by the thought of their silly chicken on the moon, so they choose that background. Now that they're ready to program the chicken, P4 lets C4 take the lead, saying, "This is the part that you're better at than me" although this is the first time for C4 has used ScratchJr. They work together to figure out which blocks to use, such as HOP, GET BIGGER, SAY HI, etc. C4 is so excited by their project and can hardly contain her laughter. After they test their program, P4 asks whether they should add another character with their chicken. Again, they scroll through all the characters, and C4 ultimately decides on a tulip and chooses not to alter the colors this time. P4 and C4 are each fully engaged in deciding which blocks to use for the tulip's program. They come up with a program, test it out, and P4 offers the suggestion to modify the tulip's program so that both the chicken and the tulip's programs take about the same amount of time. They work together to add a couple more blocks to make the programs about the same length, revising and testing as they go. C4 spontaneously begins singing a song to go along with their story of the chicken and tulip saying hi to one another. Halfway through the song, P4 encourages C4 to record the song using the RECORDED SOUND block on ScratchJr. C4 records the whole song, but something goes wrong when they try to play it back. P4 tries to attempt to problem solve while C4 gets distracted and starts moving around the room, singing her made-up song. P4 tries to get her attention by pretending she did something to mess up their program, but C4 feels "inspired to draw" a cat using construction paper and markers to go along with their ScratchJr project, remarking that the cat will not be able to catch the chicken on the moon because the tulip will defend the chicken. P4 observes her child drawing but does not fully engage in this activity. Right before the timer goes off, C4 has finished cutting out her cat and considers drawing a space chicken. During the interview, P4 commented that they had fun together but after a while, her child seemed more interested in the singing and drawing, remarking, "I think she kinda like put in the things that she liked and then she was like okay, I'm done with this."

Dyad 4's Final Project: "The Silly Chicken and the Tulip"

Dyad 5: P5 asks whether C5 would like to do an animal or something else, to which C5 responds that he wants to do an animal, specifically a tiger. Probably because C5 has extensive experience with KIBO, P5 immediately lets her son take the lead, saying, “you might have to lead the way because I don’t know how to do this.” C5 goes to the crafts table and begins looking through the different colors of construction paper. The dyad works together to find all the orange paper they can find. Seeing all the colors, C5 changes his mind and thinks of making a rainbow, but P5 encourages him to stick with the tiger idea. P5 begins drawing a face on the tiger and prompts C5 with questions about what a tiger looks like so that she can draw it properly (e.g., what color should the stripes be, does a tiger have whiskers/eyebrows/eyes/etc.). Once P5 gets the drawing started, C5 becomes more engaged and asks P5 for help in taping the tiger face to KIBO. They work together to tape the orange paper around the KIBO for the tiger’s body. When C5 expresses difficulty with using the tape, P5 teaches him how to tear the tape at an angle. The dyad bounces ideas off of each other to make the decorations sturdy and upright so that they don’t fall off the KIBO. P5 connects their activity to “putting a character on a parade float.” As the dyad continues to tape the body, C5 starts humming a song, and P5 joins him. They finish taping the tiger body to KIBO, and as a finishing touch, P5 offers a suggestion to make the tail out of a rolled-up piece of construction paper. C5 instead decides to cut a zigzag design on a rectangular strip of paper and tapes it to the back of the KIBO. Right as soon as they finish, they run out of tape, which prompts them to move to programming KIBO. P5 asks C5 for suggestions on what they should do for their program. C5 at first seems distracted but soon becomes more engaged and purposefully chooses blocks: FORWARD, BACKWARD, LIGHTS (“these are the eyes blinking”), PLAY SOUNDS. Since there are three different sounds that they can record, C5 records tiger growling sounds for the first two sounds, and P5 asks for the third turn. C5 scans the full program (containing a repeat forever loop) independently when they finish assembling, and P5 expresses her excitement to see their project come alive. After the first trial, P5 wonders what they might do differently, maybe get more tape to reinforce the decorations. During the interview, P5 remarked that when scanning their program, C5 scanned some of the blocks twice “on purpose because he wanted this to go more than once”. Whereas C5 explained that his favorite part of working with his mom was creating the tail, P5 reported that her favorite part was recording the growling sounds.

Dyad 5’s Final Project: “Tiger”

Dyad 6: C6 decides to use the play prompt and thinks that maybe he and his mom will do a ScratchJr project about dragon avengers or maybe a wizard story. After scrolling through the characters, C6 decides to choose the scenery first, settling on a theatre background since they're doing a play. The theme is "magic and mystery". As the dyad works together on changing the appearance of the wizard using the ScratchJr paint editor, they begin to call the wizard "Gandolf" and try to make the character look like this popular Lord of the Rings character. There is a lot of trial and error involved in making Gandolf. P6 shows C6 how to use the undo button, and the dyad playfully create their other Lord of the Rings characters: Frodo cat, Tree Beard, and Legolas. The dyad was unable to find a suitable existing character for Tree Beard, so C6 makes the character from scratch. C6 doesn't have enough time to create a new character for Legolas, so P6 encourages him to use the fairy instead and pretend that it's not a girl, suggested that the wings and heels are "where his bow and arrow are." When they have fewer than five minutes left, P6 asks C6 to start programming, which prompts C6 to create his own sound. C6 starts telling a story, "Once upon a time, there were these… uh hold on a second… there were these four fighters…" When he plays back the sound, C6 gets embarrassed that he messed up in the middle, to which P6 offers the suggestion of practicing what to say before recording. With only a minute left, C6 quickly makes a dance for one of the characters by stringing together several blue motion blocks. Since he has used ScratchJr before, C6 remembers to start his program with the green flag and finish with the red end block. During the post-session interview, C6 continues to add programs to the other characters so that all the characters dance together. The researcher helps the dyad to create two parallel programs so that the "Once upon a time" sound and the characters' dancing happen simultaneously.

Dyad 6's Final Project: "Lord of the Rings"

References

Ackermann, E. (2001). Piaget's Constructivism, Papert's Constructionism: What's the difference. *Future of Learning Group Publication, 5*(3), 438.

Armon, U. (1997). Cooperative Parent-Child Learning in a LEGO-Logo Environment. Retrieved from http://eurologo.web.elte.hu/lectures/armon.htm

Banerjee, R., Ko, A. J., Popovic, Z., Liu, L., Sobel, K., Pitt, C., … Yip, J. C. (2018). Empowering Families Facing English Literacy Challenges to Jointly Engage in Computer Programming. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, 1–13. https://doi.org/10.1145/3173574.3174196

Barron, B., Martin, C. K., Takeuchi, L., & Fithian, R. (2009). Parents as Learning Partners in the Development of Technological Fluency. *International Journal of Learning and Media*, *1*(2), 55–77. https://doi.org/10.1162/ijlm.2009.0021

Beals, L., & Bers, M. U. (2006). Robotic technologies: when parents put their learning ahead of their child's. *Journal of Interactive Learning Research,* 17(4), 341-366.

Bers, M. U. (2007). Project InterActions: A Multigenerational Robotic Learning Environment. *Journal of Science Education and Technology*, *16*(6), 537–552. https://doi.org/10.1007/s10956-007-9074-2

Bers, M. U. (2012). *Designing Digital Experiences for Positive Youth Development: From Playpen to Playground*. Oxford University Press, USA.

Bers, M. U. (2018). *Coding as a playground: Programming and computational thinking in the early childhood classroom.* London and New York: Routledge Press.

Bers, M. U, New, R. S., & Boudreau, L. (2004). Teaching and Learning When No One is Expert: Children and Parents Explore Technology. *Early Childhood Research and Practice, 6*(2), 1-19.

Bers, M. U., & Resnick, M. (2015). *The Official ScratchJr Book: Help Your Kids Learn to Code*. No Starch Press.

Bureau of Labor Statistics (2015). U.S. Department of Labor Employment Projections Tables. Retrieved from https://www.bls.gov/emp/tables.htm

Cheng, K. H. (2017). Exploring Parents' Conceptions of Augmented Reality Learning and Approaches to Learning by Augmented Reality with Their Children. *Journal of Educational Computing Research*, *55*(6), 820–843. https://doi.org/10.1177/0735633116686082

Clark, L. S. (2011). Parental Mediation Theory for the Digital Age. *Communication Theory*, *21*(4), 323–343. https://doi.org/10.1111/j.1468-2885.2011.01391.x

Clements, D. H., & Gullo, D. F. (1984). Effects of Computer Programming on Young Children's Cognition. *Journal of Educational Psychology*, *76*, 1051–1058.

Code.org. (2018). Landscape of CS Action in States 1 pager. Retrieved November from: https://docs.google.com/document/d/15zBdBbXUA-yEzxEq0VeWAEb9nXuGjmNFWNrYp6UdM8U/edit?usp=embed_facebook

Connell, S. L., Lauricella, A. R., & Wartella, E. (2015). Parental Co-Use of Media Technology with their Young Children in the USA. *Journal of Children and Media*, *9*(1), 5–21. https://doi.org/10.1080/17482798.2015.997440

Creswell, J. W. (2014). *Research Design: Qualitative, Quantitative and Mixed Methods Approaches* (4th edition). Thousand Oaks: SAGE Publications, Inc.

Elkin, M., Sullivan, A., & Bers, M. U. (2016). Programming with the KIBO Robotics Kit in Preschool Classrooms. *Computers in the Schools*, *33*(3), 169–186. https://doi.org/10.1080/07380569.2016.1216251

Engestrom, Y. (Ed.). (1999). *Perspectives on Activity Theory*. Cambridge; New York: Cambridge University Press.

Feng, H. C., Lin, C. H., & Liu, E. Z. F. (2011). Parents' perceptions of educational programmable bricks for kids. *British Journal of Educational Technology*, *42*(2), E30–E33. https://doi.org/10.1111/j.1467-8535.2010.01158.x

Field, A. (2009). *Discovering Statistics Using SPSS*. SAGE Publications.

Flannery, L. P., Silverman, B., Kazakoff, E. R., Bers, M. U., Bontá, P., & Resnick, M. (2013). Designing ScratchJr: support for early childhood learning through computer programming. *Proceedings of the 12th International Conference on Interaction Design and Children - IDC '13*, 1–10. https://doi.org/10.1145/2485760.2485785

Griffith, S. F., & Arnold, D. H. (2018). Home learning in the new mobile age: parent–child interactions during joint play with educational apps in the

US. *Journal of Children and Media*, 1–19. https://doi.org/10.1080/17482798.2018.1489866

Grover, S., & Pea, R. (2013). Computational Thinking in K–12: A Review of the State of the Field. *Educational Researcher*, *42*(1), 38–43. https://doi.org/10.3102/0013189X12463051

Hart, M. L. (2010). Making contact with the forgotten k-12 influence: are you smarter than your 5th grader? *Proceedings of the 41st ACM Technical Symposium on Computer Science Education - SIGCSE '10*, 254. https://doi.org/10.1145/1734263.1734349

Hiniker, A., Lee, B., Kientz, J. A., & Radesky, J. S. (2018). Let's Play!: Digital and Analog Play between Preschoolers and Parents. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, 1–13. https://doi.org/10.1145/3173574.3174233

Horn, M., & Bers, M. U. (2018). *Tangible Computing*. In *The Cambridge Handbook of Computing Education Research* (S.A. Fincher and A.V. Robins, Eds.). Cambridge University Press.

Horn, M. S., Crouser, R. J., & Bers, M. U. (2012). Tangible interaction and learning: the case for a hybrid approach. *Personal and Ubiquitous Computing*, *16*(4), 379–389. https://doi.org/10.1007/s00779-011-0404-2

Horn, M. S., Solovey, E. T., Crouser, R. J., & Jacob, R. J. K. (2009). Comparing the use of tangible and graphical programming languages for informal science education. *Proceedings of the 27th International Conference on Human Factors in Computing Systems - CHI 09*, 975. https://doi.org/10.1145/1518701.1518851

Hughes, M., & Greenhough, P. (1995). Feedback, Adult Intervention, and Peer Collaboration in Initial Logo Learning. *Cognition and Instruction*, *13*(4), 525–539.

Ivankova, N. V., Creswell, J. W., & Stick, S. L. (2006). Using Mixed-Methods Sequential Explanatory Design: From Theory to Practice. *Field Methods*, *18*(1), 3–20. https://doi.org/10.1177/1525822X05282260

Joan Ganz Cooney Center. (2014). *Zooming in: Studying family engagement with media at large and small …*. Education. Retrieved from https://www.slideshare.net/cooneycenter/aera-april2014-takeuchipressey

Kaptelinin, V., & Nardi, B. (2018). Activity Theory as a Framework for Human-Technology Interaction Research. *Mind, Culture, and Activity*, *25*(1), 3–5. https://doi.org/10.1080/10749039.2017.1393089

Kazakoff, E. R., Sullivan, A., & Bers, M. U. (2013). The Effect of a Classroom-Based Intensive Robotics and Programming Workshop on Sequencing Ability in Early Childhood. *Early Childhood Education Journal*, *41*(4), 245–255. https://doi.org/10.1007/s10643-012-0554-5

Korat, O., & Or, T. (2010). How New Technology Influences Parent—child Interaction: The Case of e-book Reading. *First Language*, *30*(2), 139–154. https://doi.org/10.1177/0142723709359242

Krcmar, M., & Cingel, D. P. (2014). Parent–Child Joint Reading in Traditional and Electronic Formats. *Media Psychology*, *17*(3), 262–281. https://doi.org/10.1080/15213269.2013.840243

Leont'ev, A. N. (1978). *Activity, Consciousness, and Personality*. Prentice-Hall Englewood Cliffs, NJ.

Lin, J. M. C., & Liu, S. F. (2012). An Investigation into Parent-Child Collaboration in Learning Computer Programming. *Journal of Educational Technology & Society*, *15*(1), 162–173.

National Association for the Education of Young Children (NAEYC), & Fred Rogers Center for Early Learning and Children's Media (2012). *Technology and Interactive Media as Tools in Early Childhood Programs Serving Children from Birth through Age 8.* Retrieved from http://www.naeyc.org/files/naeyc/ file/positions/PS_technology_WEB2.pdf

National Early Literacy Panel. (2008). *Developing Early Literacy: Report of the National Early Literacy Panel*. Retrieved from https://lincs.ed.gov/publications/pdf/NELPReport09.pdf

Nederhof, A. J. (1985). Methods of coping with social desirability bias: A review. *European Journal of Social Psychology*, *15*(3), 263–280. https://doi.org/10.1002/ejsp.2420150303

Papert, S. (1980). *Mindstorms: children, computers, and powerful ideas*. New York: Basic Books.

Parish-Morris, J., Mahajan, N., Hirsh-Pasek, K., Golinkoff, R. M., & Collins, M. F. (2013). Once Upon a Time: Parent–Child Dialogue and Storybook Reading in the Electronic Era. *Mind, Brain, and Education*, *7*(3), 200–211. https://doi.org/10.1111/mbe.12028

Pearce, J., & Borba, S. (2017). What Is Family Code Night? Retrieved November 25, 2018, from NAESP website: https://www.naesp.org/blog/what-family-code-night

Pierson, E., Momoh, L., & Hupert, N. (2015). *Summative Evaluation Report for the Be A Scientist! Project's Family Science Program*.

Portelance, D. J., Strawhacker, A. L., & Bers, M. U. (2016). Constructing the ScratchJr programming language in the early childhood classroom. *International Journal of Technology and Design Education*, *26*(4), 489–504. https://doi.org/10.1007/s10798-015-9325-0

Pretz, K. (2014). Computer Science Classes for Kids Becoming Mandatory. Retrieved from http://theinstitute.ieee.org/career-and-education/education/computer-science-classes-for-kids-becoming-mandatory

Pugnali, A., Sullivan, A., & Umashi Bers, M. (2017). The Impact of User Interface on Young Children's Computational Thinking. *Journal of Information Technology Education: Innovations in Practice*, *16*, 171–193. https://doi.org/10.28945/3768

Resnick, M., & Silverman, B. (2005). Some reflections on designing construction kits for kids. *Proceeding of the 2005 Conference on Interaction Design and Children - IDC '05*, 117–122. https://doi.org/10.1145/1109540.1109556

Rideout, V. J. (2014). *Learning at home: Families' educational media use in America.* A report of the Families and Media Project. New York: The Joan Ganz Cooney Center at Sesame Workshop.

Rideout, V. (2017). *The Common Sense Census: Media use by kids age zero to eight.* San Francisco, CA: Common Sense Media.

Roque, R. (2016). Family Creative Learning: Designing Structures to Engage Kids and Parents as Computational Creators. In Peppler, K., Kafai, Y., & Halverson, E. (Eds.) *Makeology in K-12, Higher, and Informal Education*. New York, NY: Routledge.

Roque, R., Lin, K., & Liuzzi, R. (2014). Engaging Parents as Creative Learning Partners in Computing, *Exploring the Material Conditions of Learning, 2,* 687-688.

Roque, R., Lin, K., & Liuzzi, R. (2016). "I'm Not Just a Mom": Parents Developing Multiple Roles in Creative Computing. In *Proceedings of the 12th International Conference of the Learning Sciences*, 1-8. Singapore.

Sanford, C., Knutson, K., & Crowley, K. (2007). "We Always Spend Time Together on Sundays": How Grandparents and Their Grandchildren Think About and Use Informal Learning Spaces. *Visitor Studies*, *10*(2), 136–151. https://doi.org/10.1080/10645570701585129

Sapounidis, T., & Demetriadis, S. (2013). Tangible versus graphical user interfaces for robot programming: exploring cross-age children's preferences. *Personal and Ubiquitous Computing*, *17*(8), 1775–1786. https://doi.org/10.1007/s00779-013-0641-7

Strawhacker, A., & Bers, M. U. (2015). "I want my robot to look for food": Comparing Kindergartner's programming comprehension using tangible, graphic, and hybrid user interfaces. *International Journal of Technology and Design Education*, *25*(3), 293–319. https://doi.org/10.1007/s10798-014-9287-7

Strawhacker, A., & Bers, M. U. (2018). What they learn when they learn coding: investigating cognitive domains and computer programming knowledge in young children. *Educational Technology Research and Development*. https://doi.org/10.1007/s11423-018-9622-x

Strawhacker, A., Sullivan, A., & Bers, M. U. (2013). TUI, GUI, HUI: is a bimodal interface truly worth the sum of its parts? *Proceedings of the 12th International Conference on Interaction Design and Children - IDC '13*, 309–312. https://doi.org/10.1145/2485760.2485825

Sullivan, A., Bers, M. U., & Mihm, C. (2017). Imagining, Playing, and Coding with KIBO: Using Robotics to Foster Computational Thinking in Young Children. In *Proceedings of the International Conference on Computational Thinking Education.* Wanchai, Hong Kong.

Swartz, M. I., & Crowley, K. (2004). Parent Beliefs about Teaching and Learning in a Children's Museum, *Visitor Studies,* 7(2), 5-16.

Takeuchi, L., & Stevens, R. (2011). *The New Coviewing: Designing for Learning through Joint Media Engagement.* The Joan Ganz Cooney Center at Sesame Workshop. New York.

Tucker, L., Scherr, R. E., Zickler, T., & Mazur, E. (2016). Exclusively visual analysis of classroom group interactions. *Physical Review Physics Education Research*, *12*(2). https://doi.org/10.1103/PhysRevPhysEducRes.12.020142

Vygotsky, L. S. (1980). *Mind in Society*. Harvard University Press.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35.

Wing, J. M., & Stanzione, D. (2016). Progress in computational thinking and expanding the HPC community. *Communications of the ACM*, *59*(7), 10–11. https://doi.org/10.1145/2933410

Xie, L., Antle, A. N., & Motamedi, N. (2008). Are tangibles more fun?: comparing children's enjoyment and engagement using physical, graphical and tangible user interfaces. *Proceedings of the 2nd International Conference on Tangible and Embedded Interaction - TEI '08*, 191. https://doi.org/10.1145/1347390.1347433

Yu, J., & Roque, R. (2018). A survey of computational kits for young children. *Proceedings of the 17th ACM Conference on Interaction Design and Children - IDC '18*, 289–299. https://doi.org/10.1145/3202185.3202738