Visitor Studies

ISSN: 1064-5578 (Print) 1934-7715 (Online) Journal homepage: https://www.tandfonline.com/loi/uvst20

Engaging Children and Parents to Code Together Using the ScratchJr App

Madhu Govind, Emily Relkin & Marina Umaschi Bers

To cite this article: Madhu Govind, Emily Relkin & Marina Umaschi Bers (2020) Engaging Children and Parents to Code Together Using the ScratchJr App, Visitor Studies, 23:1, 46-65, DOI: 10.1080/10645578.2020.1732184

To link to this article: https://doi.org/10.1080/10645578.2020.1732184

Published online: 30 Mar 2020.

VISITOR STUDIES
2020, VOL. 23, NO. 1, 46–65
https://doi.org/10.1080/10645578.2020.1732184

Engaging Children and Parents to Code Together Using the ScratchJr App

Madhu Govind, Emily Relkin, and Marina Umaschi Bers

Tufts University, Medford, Massachusetts, USA

ABSTRACT
Educational tools and apps designed to teach coding and computational thinking to children have risen in popularity in the last several years. However, there is little research that explores how families with young children code together in informal environments. This study explored how children ages 5-7 and their parents jointly program with the ScratchJr app. $N = 58$ families attended ScratchJr Family Days, single-day events for families to engage in an interactive ScratchJr session. Three additional parent-child dyads participated in a follow-up observational study, in which they were videotaped while working on an open-ended ScratchJr activity. Findings indicated that parents reported engaging as observers and coaches, whereas children engaged as planners. There were moderate, positive associations between children's prior coding interest and their engagement in debugging, as well as between children's role as playmates and their engagement in the design process. Implications and opportunities for future research are discussed.

Many technologies for young children in the twenty-first century have focused on introducing them to coding and computational thinking. Platforms include and are not limited to robotic kits, video and computer games, and tablet-based games and coding apps (Code.org, 2019; Yu & Roque, 2018). These technologies 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 to introducing computer science to young children (Bers, 2018a; Clements & Gullo, 1984; Strawhacker & Bers, 2019). These advancements have led to policy changes at national and international levels. As of December 2019, 34 states in the United States have adopted K-12 computer science frameworks and standards (Code.org, 2019), and countries such as Finland, United Kingdom, and Estonia have begun embedding computer science into their elementary educational frameworks (Pretz, 2014; UK Department for Education 2013).

Alongside the increasing popularity of coding technologies and computer science education is the rise in young children's technology and media use. Although young children's screen use is an ongoing issue for many parents and educators of young children, the American Academy of Pediatrics (AAP) modified its policy recommendations from restricting children's screen-based technology usage to instead encouraging parents to co-use media with their children (Connell, Lauricella & Wartella, 2015). Referred to as Joint Media Engagement (JME) in the literature, researchers seek to better understand how the contexts in which technologies are shared among multiple individuals can enhance children's learning and development (Takeuchi & Stevens, 2011).

Parents play an important role in engaging their children in technology-mediated activities, including those that involve coding. Various models of family coding events, during which children and parents are invited to create projects or play with coding software together, have emerged in recent years such as Family Code Night (Pearce & Borba, 2017) and Family Creative Learning (Roque, 2016; Roque, Liu, & Liuzzi, 2015; Roque, Liu, & Liuzzi, 2016), and have demonstrated positive engagement in coding among families. However, these models primarily focus on children in late elementary school or older. There is a gap in the current literature on family coding in early childhood, a time period of children's development that parents believe is crucial for fostering their children's interest in coding. In fact, a national survey of 2,000 parents revealed that 85% of parents consider coding to be a valuable skill set for their young children, with the ideal age for introducing coding toys being between six and seven years old. In the same survey, however, 72% of parents indicated that children's better understanding of technology makes it more difficult for parents to help their children learn (The Toy Association, 2017). These findings highlight the importance of understanding parent perceptions of the possible benefits of family-oriented coding.

This paper discusses two research studies that explored how young children and parents co-engaged with the ScratchJr programing application. In the first study, parents attended informal coding workshops called "ScratchJr Family Days" with their children and reported their experiences in pre- and post-surveys. The primary research questions for this first study were as follows: What are parent perceptions of the roles assumed by parents versus children during joint coding with ScratchJr? What are the relationships among children's coding interest, role engagement, and CT engagement (as reported by their parents)? We then conducted a second follow-up study, "Parent-Child Interactions with ScratchJr", with three families to further illuminate survey findings and to describe how parents and young children code together with the ScratchJr app. The primary research question for this second study was as follows: What are qualitative examples of roles assumed by children and parents, as well as children's CT engagement, during joint coding with ScratchJr?

Related work

Prior research has suggested that parents assume a variety of roles when jointly engaging with technology with their children. 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. These roles illustrate how parents not only offer support for their children but also engage in learning themselves through legitimate peripheral participation (Barron et al., 2009; Lave & Wenger, 1991). Similarly, parents and grandparents who visited interactive museum exhibits with their children were found to have engaged in three distinct roles: teacher, by instructing children to attend to different features of the exhibit; coach, by encouraging their children' learning of new things; and playmate, by learning and playing alongside their children (Sanford, Knutson & Crowley, 2007; Swartz & Crowley, 2004).

In the context of coding, studies have shown that parents with little to no background in programing or technology tended to allow their child to be the driver while they took on more passive roles such as reviewer or novice, though some parents found it challenging to be less knowledgeable than their children (Lin & Liu, 2012; Roque, Lin, & Liuzzi, 2016). Conversely, parents with an information technology (IT) background showed higher competence and confidence when working with their children. Because of this "expertise", however, parents were less likely to learn from their mistakes and to let children explore and tinker (Bers, 2007; Bers, New, & Boudreau, 2004; Feng et al., 2011). Less time for tinkering has been associated with decreased student performance (Beals & Bers, 2006; Hughes & Greenhough, 1995) even if parent-child joint projects tended to be more complex and involved children using more systematic methods of problem solving rather than simply trial-and-error (Lin & Liu, 2012).

Participation in coding activities gives children an opportunity to learn by enhancing their computational thinking (CT) skills. Although the term computational thinking has many definitions, in its contemporary form CT is thought to be a broad set of thought processes used for problem solving and thinking, not just in the way that humans program computers, but in ways that humans think (Papert, 1980; Wing, 2006). The IEA International Computer and Information Literacy Study defines CT as an "individual's ability to recognize aspects of real-world problems that are appropriate for computational formulation and to evaluate and develop algorithmic solutions to those problems so that the solutions can be operationalized with a computer" (Fraillon, Ainley, Schulz, Duckworth, & Friedman, 2018). The Computer Science Teachers Association (CSTA) categorizes CT into five categories: data representation, decomposition, abstraction, algorithmic thinking, and patterns (International Society for Technology in Education and the Computer Science Teachers Association, 2011). Bers (2018) identified seven powerful ideas of computational thinking that are developmentally appropriate for early childhood. These ideas include algorithms, modularity, control structures, representation, hardware/software, design process, and debugging. Studies show that children, even as young as four, can grasp CT topics such as algorithmic thinking, the iterative design process, and debugging strategies (Bers, 2018a; Clements & Gullo, 1984).

In this paper we describe findings from two studies that explored family-oriented programing with ScratchJr, a free tablet-based coding app for young children (Bers & Resnick, 2015; Flannery et al., 2013). ScratchJr was developed through National Science Foundation funding as a research collaboration among the DevTech Research Group at Tufts University, the Lifelong Kindergarten Group at the MIT Media Lab, and the Playful Invention Company. Using ScratchJr children can design and animate characters using 20 different types of block-based programing blocks. Figure 1 displays the ScratchJr interface. The ScratchJr app was designed with a "low floor, high ceiling, wide walls" approach so that individuals of diverse levels of experience can tinker with the graphical programing blocks to create imaginative stories and games (Bers, 2018; Portelance, Strawhacker & Bers, 2019; Resnick & Silverman, 2005).

Figure 1. ScratchJr interface.

This paper, which describes findings from the ScratchJr Family Days and the Parent-Child Interactions with ScratchJr studies, builds upon the prior work in several ways. Firstly, we focused specifically on family-oriented coding with young children (ages 5-7) in order to fill the early childhood gap in the existing literature. Secondly, this paper discusses two different but complementary studies: one utilizing a community-oriented approach that took place in informal settings and the other utilizing an experimental approach that took place in a lab setting. Through the former approach, we focused on parent beliefs regarding their role dynamics and how these roles might be related to children's coding interest and CT engagement. Through the latter, we identified examples of how these roles were enacted and how children exhibited various CT skills. Taken together, both studies provided insight into how children and families code together using the ScratchJr app.

Method

Study 1: ScratchJr Family Days

Procedure

We first piloted ScratchJr Family Day events at several local schools and museums as part of a community outreach initiative to promote family engagement. Feedback from these events suggested that families enjoyed learning and creating projects together with ScratchJr, and that other individuals were interested in hosting family-oriented events in their respective communities (e.g., after-school programs, museums, community centers, etc.). We thus developed a ScratchJr Family Day Protocol, which outlined the process of

Table 1. ScratchJr Family Day protocol.

	1-hour event	1.5-hour event	2-hour event
Arrival and check-in: Families sign in and check out tablets if they did not bring their own	5 min	10 min	15 min
Introduction to ScratchJr: Children engage in off-screen activities and learn about programing and the ScratchJr interface; parents complete pre-survey and learn about ScratchJr through hands-on play and the tip sheet	15 min	20 min	30 min
Family coding: Children and parents co-create a ScratchJr project using one of the activity prompts as inspiration	25 min	20 min	20 min
Family swap and share time: Families pair up and give feedback on projects		5 min	10 min
Family coding: Children and parents use feedback to make changes to their projects		10 min	15 min
Group share: Families share their projects with the whole group	10 min	15 min	20 min
Clean up: Families clean up materials and return borrowed tablets	5 min	10 min	10 min

hosting a ScratchJr Family Day and included sample recruitment materials, agendas, and resources for facilitators. The protocol and other resources used in this research were approved by the university's Institutional Review Board. Agendas ranged from 1-2 hours and provided space for children and adults to engage separately with ScratchJr before coming together for the project. This model was adapted from the Family Creative Learning workshops (Roque, Lin & Liuzzi, 2016) to ensure that children and parents could develop their own ideas before becoming collaborators.

Table 1 summarizes the general ScratchJr Family Day protocol. Three different activity prompts were provided for the Family Coding activity, but families and facilitators were encouraged to use these prompts or create their own project themes: program a character in ScratchJr to perform your favorite dance (ScratchJr Dance); program your favorite animal in ScratchJr and design its habitat (ScratchJr Animal); or program a character in ScratchJr to act out a scene from your favorite movie or book (ScratchJr Play).

Study recruitment

Recruitment for ScratchJr Family Day events was two-fold through various e-list and social media platforms: families were recruited to attend events hosted by DevTech at Tufts University, and facilitators around the country were recruited to host events for families in their community. Facilitators could be teachers, parents, community members, or anyone interested in bringing families together to code. ScratchJr Family Day events were promoted as free and fun educational experiences for families to learn about ScratchJr and work together on a creative project. Research participation was voluntary and consisted of pre- and post-surveys completed by parents/legal guardians. Facilitators could also opt to participate in research by completing a post-survey within 48 hours after hosting a ScratchJr Family Day. For the purposes of highlighting parent perceptions of ScratchJr Family Days, facilitator survey responses are not presented in this paper. Surveys consisted of multiple-choice, open-ended, and Likert-style type questions. Personal identifying information consisted of parent and facilitator names, which were collected in order to pair surveys at different time points. Once pairing was done, the names were removed securely, and participants were randomly assigned a unique identification number.

Families and facilitators were recruited between October 2017 and July 2018, with pilot events occurring prior to October 2017. Pilot data are not presented nor analyzed in this paper. In total 142 facilitators expressed interest in hosting a ScratchJr Family Day during this time period. Nine events were conducted (two DevTech-hosted and seven facilitator-hosted), and findings from these events are presented in this paper. Independent sample *t*-tests revealed no significant demographic differences between the families who chose to participate in DevTech-hosted events versus facilitator-hosted events (all p's $> .05$). Seventy pre-surveys and 63 post-surveys were completed by parents/legal guardians (henceforth referred to as simply parents), with $N = 58$ parents completing both pre-and post-surveys. One of the 70 participants was not a parent/guardian and thus did not sign informed consent. One potential reason for missing survey data is that families may have arrived late or needed to leave early, thus being unable to complete both surveys. Another possible reason is that parents used different identifiers when completing the two surveys, in which case their responses could not be paired together. Due to the nature of data collection at external sites, it is unclear whether all families who attended an event also chose to participate in the research surveys. However, the number of surveys collected at the events matches the approximate ranges of families in attendance reported in the facilitator surveys.

Participants

The analytic sample for this first study is the $N = 58$ parents who completed both pre- and post-surveys. Although younger and older siblings (ages 4-13) and other family members were encouraged to attend and participate, only one parent from each family completed surveys. Each parent was asked to only report on experiences with their children ages 5-7 ($M_{age} = 6.4$ years, $SD = 1.18$). Almost all families attended with 1-2 parents (92%) and 1-2 children (100%). Forty-four parents were mothers, and 14 were fathers. Parents were highly educated, with 80% holding at least a master's degree. Forty-three percent of parents reported being in a STEM-related profession (e.g., software or other branch of engineering, information technology, etc.); those not working in a STEM-related field gave examples of working in law, business, or healthcare. About half of parents (53%) and children (45%) were reported to have never coded or identified as beginners; 47% of parents reported that their children had previously used ScratchJr.

Study 2: Parent–child interactions

Procedure

In order to explore families' interactions with ScratchJr at a closer level, we conducted case studies of parent-child dyads engaging with ScratchJr in a closed experimental setting. Dyads were individually recruited and invited into a testing room connected to a one-way-view observation booth. The parent first completed a brief pre-survey while the researcher allowed the child to freely explore ScratchJr. The dyad was then given

20 minutes to work together on a ScratchJr project. Sample prompts from the ScratchJr Family Days protocol (Animal or Play) were provided, or the dyad could come up with their own idea for their project. One or sometimes two researchers observed from the observation booth and recorded field notes. If the parent or child needed help at any point, they were to step outside the testing room, and a researcher would come and assist them (as a facilitator would do at a ScratchJr Family Day).

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 programing experience. A semi-structured approach allowed the researcher to probe and ask specific questions that would elicit a greater understanding of their interactions and role engagement. For example, the researcher asked broad questions such as "Tell me about your project" and more specific questions about the tasks or ideas that each person contributed to the programing activity. After the interview, parents completed a brief post-survey. The entire coding play session and semi-structured interview were videotaped and transcribed. Though the survey items were similar to those in the first study, survey responses from the three parents are not presented in this paper.

Study recruitment

Families were recruited through the DevTech e-list and word-of-mouth. Inclusion criteria for participation were that the child must be between five and seven years old, the parent must be able to complete surveys, and both the child and parent must have been able to converse in English for the duration of the study.

Participants

Three parent-child dyads participated in the study. The three families were demographically similar to the first study's sample, though none of the three dyads had previously participated in a ScratchJr Family Day. All names have been replaced with pseudonyms to maintain participant confidentiality.

The first dyad included Andrew (age 6), who had no prior coding experience, and his mother Pamela (age 41), who worked in a non-STEM profession, held a bachelor's degree, and had previously used ScratchJr in a limited capacity. The second dyad included Dani (age 7), who had extensive experience with other coding platforms such as the KIBO robotics kit but had never used ScratchJr, and her mother Lara (age 37), who worked in a non-STEM profession, held a bachelor's degree, and also had engaged with KIBO but never ScratchJr. The third dyad included Brendan (age 6), who had extensive experience with ScratchJr at a prior summer camp, and his mother Olivia (age 37), who worked in a non-STEM profession, held a master's degree, and had no prior coding experience.

We acknowledge that a few case studies could never be truly representative of every parent-child dyad who had attended a ScratchJr Family Day or had co-engaged with ScratchJr through other avenues. However, these case studies were still meaningful because they provided rich qualitative examples of how parent-child dyads enacted in different roles and how their children exhibited various CT skills during their interactions with the ScratchJr programing application.

Constructs

Role engagement

Parent and child role engagement was measured in the parent post-event survey using predetermined role categories from Sanford, Knutson, & Crowley (2007) and Barron and colleagues (2009). Of the role categories in this aforementioned study, the following five roles were chosen based on their relevance to the activity of programing: planner (planned out project topic and delegated tasks to members of the group), observer (let others guide project creation, did not contribute to the group's coding activities), teacher (explained some of the coding topics to the group during the activity), coach (encouraged and supported the group, offered suggestions to group members during the activity), and playmate (shared the fun, enjoyable parts of the activity with the group). Parents were asked to rate the extent to which they engaged in each role on 1-5 Likert-type scales (5 being the highest). In addition to the parent survey items, qualitative examples from the Parent-Child Interactions study brought to light the specific behaviors and actions that constituted the various parent and child roles.

Coding interest

In Study 1, both children and parents' level of coding interest were measured in the parent pre- and post-survey on 1-5 Likert-type scales (5 being the highest). Coding interest was not a focus in Study 2.

Computational thinking skills

We surveyed the extent to which children displayed algorithmic thinking, the iterative design process, and debugging because these categories of CT are promoted by the ScratchJr programing application and are readily observed in coding sessions (Bers, 2018b). In the context of ScratchJr, children use algorithms when sequencing programing blocks and choosing the order to the various subroutines of their programs. Children use the design process when iteratively planning and testing programs to achieve the goals of their project. Debugging occurs when children identify problems with their code and identify strategies to fix them.

In Study 1, parents reported on the extent to which children displayed these three computational skills and concepts: algorithms (coding with a logical sequence of steps), design process (ask, imagine, plan, create, test and improve, share), and debugging (finding and correcting errors, troubleshooting). Children's engagement in CT skills was measured in the parent post-survey on a 1-5 Likert-type scale (5 being the highest). For example, a score of 5 meant that the "child correctly troubleshoots the issue and fixes the problem". In addition, qualitative examples from the Parent-Child Interactions study illuminated the specific ways children exhibited the three CT skills using ScratchJr in the context of joint family programing.

Data analysis

Study 1: ScratchJr Family Days

SPSS Statistics Version 25 was used to analyze the $N = 58$ matched survey responses. There were no significant differences in the responses from the 12 participants who filled out only the pre-survey and the 58 participants who completed both surveys. The five participants who only completed the post-survey had significantly higher responses on several survey items as compared to the 58 participants (e.g., child and parent roles and children's CT engagement). However, this was a very small subset of the sample, so it is not likely that these differences are meaningful. Data were screened to check for normality, homogeneity of variance, outliers, and other assumptions underlying the subsequent analyses.

Our first research question of ScratchJr Family Days was: What are parent perceptions of the roles assumed by parents versus children? To answer this question, we conducted independent sample *t*-tests to explore whether there were any differences in parents' self-reported roles versus children's reported roles. The Bonferroni correction was applied to account for the risk of Type I error when running multiple comparison tests, resulting in an adjusted p value of $.05/5 = .01$ for determining statistical significance.

Our second research question was: What are the relationships among children's coding interest, role engagement, and CT engagement as reported by their parents? In other words, would parents who believed their children were highly interested in coding also believe that their children engaged more deeply in particular roles or CT skills during the joint ScratchJr activity? We also sought to explore if there was a relationship between specific CT domains and roles that children were reported to engage in. To answer these questions, we computed Pearson product-moment correlation coefficients to assess the relationships among these nine variables: child's coding interest from the parent pre-survey, the five child roles (planner, observer, teacher, coach, playmate) from the parent post-survey, and the three CT skills (algorithms, design process, debugging) from the parent post-survey.

Study 2: Parent–child interactions

Qualitative data from the videotaped observations and 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 the researchers in coding the transcripts. In order to answer the research question, "What are qualitative examples of roles assumed by children and parents, as well as children's CT engagement?", we deductively coded the transcripts for the five parent and child roles and the three CT skills. Discrepancies were resolved by discussing the codes as a team and reaching agreement. Examples of how these roles were enacted by the three parent-child dyads and how children in these case studies exhibited the CT skills are provided in order to illuminate the quantitative findings from the first study.

Results

Study 1: ScratchJr Family Days

Table 2 displays the means and standard deviations of parents' ratings of their own and their children's role engagement during ScratchJr Family Day events. Independent

Table 2. Parent- and child-reported role engagement during ScratchJr Family Days.

Role	Child mean *(SD)*	Parent mean *(SD)*	Absolute value of mean difference	*t*	*p*
Planner	4.31 (1.10)	2.70 (1.42)	1.61	6.59	.000
Observer	2.52 (1.49)	3.74 (1.24)	1.22	4.54	.000
Teacher	3.09 (1.47)	3.37 (1.44)	0.27	0.95	.343
Coach	3.09 (1.46)	4.20 (0.98)	1.10	4.54	.000
Playmate	4.46 (0.91)	4.31 (0.91)	0.15	0.85	.398

Table 3. Correlation table of children's coding interest, roles, and CT engagement.

	Coding interest (1)	Planner (2)	Observer (3)	Teacher (4)	Coach (5)	Playmate (6)	Algorithms (7)	Design process (8)	Debugging (9)
1	1	.25	.07	.42**	.32*	.32*	.37**	.38**	.56**
2		1	−.04	.27	.30*	.14	.26	.41**	.20
3			1	.55**	.53**	.16	.18	−.02	.01
4				1	.86**	.41**	.37**	.37**	.35**
5					1	.42**	.33*	.36*	.23
6						1	.38**	.51**	.48**

**Significant at the .01 level (2-tailed).
*Significant at the .05 level (1-tailed).

sample *t*-tests revealed that parents reported engaging significantly more as coaches, $t(91.36) = 4.54$, $p < .001$, $d = 0.89$, and as observers, $t(99.14) = 4.54$, $p < .001$, $d = 0.89$, as compared to their children. Children were reported to have engaged significantly more as planners, $t(99.52) = 6.59$, $p < .001$, $d = 1.27$, as compared to their parents. These large effect sizes indicate that parents reported engaging in distinctly separate roles from their children during the joint coding activity.

Table 3 displays the Pearson product-moment correlation coefficients, which quantify the strength of relationships among children's reported coding interest prior to attending ScratchJr Family Day, their reported engagement in the five roles (planner, observer, teacher, coach, playmate), and their reported engagement in the three CT skills (algorithms, design process, debugging) during the collaborative ScratchJr activity. Children's reported coding interest was weakly and positively associated with the teacher ($r = .42$, $p = .002$), coach ($r = .32$, $p = .020$), and playmate roles ($r = .32$, $p = .019$), as well as with children's engagement in algorithmic thinking ($r = .37$, $p < .007$) and the design process ($r = .38$, $p = .005$). There was a moderate, positive association between children's reported coding interest and engagement in debugging, $r = .56$, $p < .001$. The teacher and playmate roles were both weak-to-moderately associated with all three CT skills. The planner role was weakly and positively associated with engagement in the design process, $r = .41$, $p = .002$. There was a weak, positive association between the coach role and algorithms ($r = .33$, $p = .016$), as well as between the coach role and design process ($r = .36$, $p = .008$).

Study 2: Parent–child interactions

In order to ground the quantitative findings and provide a snapshot of how children and parents enacted these roles and how children exhibited the three CT skills, we next provide examples from the three case studies.

Figure 2. Dyad 1's final ScratchJr project.

Andrew (child) and Pamela (parent)

The play session began with Andrew choosing characters for their project: a wizard, a seahorse and later, a fairy. Pamela observed and encouraged Andrew to start programing his characters and allowed him to take the lead.

> *Pamela: "Can you show me how to do actions because I didn't really catch that. I haven't really done this before."*
>
> …
>
> *Andrew: "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"*

The dyad talked through how they would first program the fairy to "do magic" by programing the fairy to move backwards for six steps, make a "pop" sound, shake, and then make a "whoop whoop" sound (see Figure 2). Once the dyad had a complete program, they tested it out. Pamela asked if Andrew wanted to add a different character, but they decided that they would instead program the wizard next.

> *Andrew: "Should we make it repeat all over and over again, or should we end it?"*
>
> *Pamela: "What do you think?"*
>
> *Andrew: "Repeat all over and over. Let's see what that does."*
>
> …
>
> *Pamela: "Why don't you come up with something he should say? Your ideas are so original."*

As they continued to make changes to the wizard's program, Pamela accidentally swiped up the program, which deleted their entire code. Andrew became upset, but Pamela quickly assured him that she remembered their program. She took control of the tablet as they discussed the blocks that they would keep the same as before and which ones they would now modify.

Toward the end of their session, Andrew asked to include a new character, maybe one where he would include his own face. Not knowing how to do this, Pamela called the researcher for help. The researcher entered the room and helped Andrew insert his face inside an astronaut character.

Together, Andrew and Pamela programed the astronaut to move up and down and say "Hi". Figure 2 displays Andrew and Pamela's final project and the program for their fairy character.

In this case study, Pamela enacted the roles of observer and coach by guiding the project forward and prompting her son to think about the larger story with the seahorse, fairy, and wizard characters. Andrew enacted the role of planner by getting to choose the characters and deciding the specific blocks used for their codes. He tested the program after each character was programed, which afforded him the opportunity to utilize the iterative design process of planning, creating, testing and improving. Furthermore, when Pamela allowed her son to take the lead, Andrew was able to showcase his knowledge by teaching his mother how to modify the number of steps their character would take. However, Andrew did not have an opportunity to debug. When one of their character's programs was accidentally deleted halfway through their project, Andrew recognized that something was wrong, but Pamela took control of the tablet, attempting to recreate their program. Rather than debugging together, Pamela used this opportunity to encourage Andrew to revise and improve upon their code; as a result, Andrew was able to engage in algorithmic thinking and in the design process.

Dani (child) and Lara (parent)

Their play session began by Lara reading aloud the two activity prompts: Animal and Play. Dani immediately chose the play prompt but did not have an idea in mind. With Lara's encouragement to scroll through all the different ScratchJr characters, Dani finally decided on a chicken. Lara encouraged Dani's playful behaviors when choosing a background for their chicken character.

> *Lara: "Is it a nighttime chicken or a schoolhouse chicken? Is it a savannah chicken or a SPACE chicken? (Dani laughs) maybe it is! Space chicken! Check!"*

When the dyad began to program their "space chicken", Lara let Dani plan it out, claiming, "This is the part that you're better at than me" although this session was also the first time that Dani had ever used ScratchJr. The dyad worked together to figure out which blocks to use, such as HOP, GET BIGGER, SAY HI, etc. (see Figure 3). Dani could hardly contain her laughter when they tested out the program. Dani chose a new character to add to their project—a tulip—and began programing it. Lara offered the suggestion to modify the tulip's program so that both the chicken and the tulip's programs ran for the same amount of time. Dani and Lara worked together to add more blocks to make the programs the same length, revising and testing as they went along.

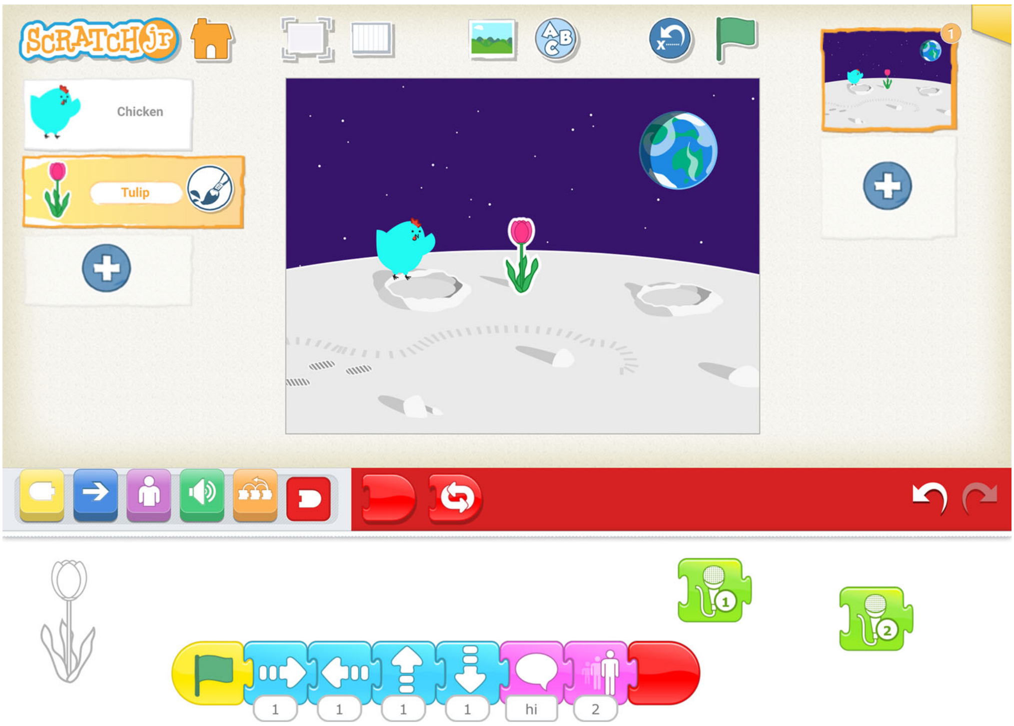

Figure 3. Dyad 2's final ScratchJr project.

In the middle of their session, Dani spontaneously began singing a song about their ScratchJr program. Lara laughed and encouraged her to record the song using the RECORD SOUND block. However, they had trouble recording the song and tried to figure out what was wrong. After three unsuccessful tries, they decided to leave their recorded sound out of their final code (see Figure 3), and Dani instead felt "inspired to draw" a cat on construction paper.

For the remainder of the session, Lara observed her child drawing and encouraged Dani's creative storytelling.

Dani: "My cat is happy, see?"

Lara: "I bet she'll be really happy if she catches that chicken on the moon."

Dani: "Bet she'll not be happy with that chicken on the moon because I have a tulip as a friend."

Lara: "Oh! You think the tulip will defend the chicken against the cat?"

In this case study, Dani and Lara exemplified the role of playmates as they bounced ideas off one another and encouraged each other's creativity and silliness. Rather than serving as a passive observer, Lara asked questions, engaged in dialogue, and redirected Dani's attention whenever she got too distracted with singing or drawing. Dani took control of clicking the various icons on the ScratchJr interface while her mother held the tablet, which allowed for Dani to assume the planner role and make decisions about their project. Although Dani and Lara were unable to figure out how to record the

Figure 4. Dyad 3's final ScratchJr project.

song, Dani was eager to try different debugging strategies, such as removing the block and re-adding it to the program, re-recording the sound, and trying to record a shorter sound.

Brendan (child) and Olivia (parent)

Brendan, having had a lot of experience with ScratchJr already, immediately decided that they would use the Play prompt, and that he and his mother Olivia would either create a project about "dragon avengers" or a wizard story. Regardless, their theme would be "magic and mystery". They settled on using the popular Lord of the Rings characters and worked together on editing the appearance of "Gandolf, Frodo cat, Treebeard, and Legolas" using the ScratchJr paint editor. Most of Brendan and Olivia's coding session did not involve coding at all; their time was mostly spent designing their characters and personalizing their characters' names (see Figure 4).

Olivia: "We gotta get rid of the wings."

Brendan: "And the tutu. Also those shoes! Legolas does not wear those!"

Olivia: "Well we can pretend he does… Let's just pretend that's where his bow and arrow are."

Brendan: "Okay."

When they had fewer than five minutes left, Olivia asked Brendan to start programming their characters. Brendan used the RECORD SOUND block to tell a story about their project.

Brendan: "Once upon a time, there were these… uh hold on a second… there were these four fighters… (records full story but messes up in the middle) I'm so embarrassed now!"

Olivia: 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."

With only a few minutes remaining, Brendan strung together several blue motion blocks swiftly with ease. Olivia observed as Brendan assembled his programs using proper ScratchJr syntax by beginning his codes with the BEGIN ON GREEN FLAG block and ending his codes with the END block (see Figure 4).

Being an experienced ScratchJr user, Brendan often assumed the teacher role and explained to his mother how to navigate the app. The dyad worked collaboratively as playmates as they exchanged ideas on how to design their characters and took turns coloring in their characters' outfits in the ScratchJr paint editor. As a coach, Olivia offered suggestions and provided words of encouragement, especially when Brendan realized he made a mistake. Brendan primarily exhibited the design process in the creation of his ScratchJr characters. His engagement in algorithmic thinking was evident toward the end of the coding session when he was able to quickly and accurately assemble his program. However, because Brendan only used the programing blocks for a few minutes, we found no evidence of debugging during their interaction.

Discussion

In this paper we describe two studies that focused on how young children and families can jointly engage in creative coding activities using ScratchJr, a free block-based programing application targeted for early childhood. Study 1, ScratchJr Family Days, explored parent perceptions of the various roles assumed by children and parents. In order to better understand participants' perceptions of what happened during the event (i.e., the roles they played) and their children's interest in coding and learning outcomes, we also explored the relationships among children's reported coding interest, role engagement, and CT engagement as reported by parents. The second study, Parent-Child Interactions, provided qualitative examples of how these roles were enacted and how children exhibited various CT skills in the context of working collaboratively with their parents on an open-ended ScratchJr project. We discuss the findings in tandem, using the qualitative examples from the second study to augment the quantitative findings from the former.

We first sought to explore parent perceptions of the roles assumed by parents versus children during ScratchJr Family Days. Findings suggested that parents reported engaging more as observers and coaches, whereas they reported their children to engage more as planners. These distinctions support previous findings by Lin and Liu (2012) who found that when parent-child dyads engage in pair programing, the child tends to assume the "driver" role and leads the project, whereas the parent assumes a more passive role such as "reviewer". Particularly in early childhood, parents seek to facilitate their children's learning and offer support by providing verbal or physical cues, rather than taking full ownership of a project or utilizing an instructional approach.

These behaviors were present in the three qualitative case studies, in which mothers would provide words of encouragement, ask prompting questions, or point to icons on the ScratchJr interface to help guide their children's interactions with the app. Other studies involving parent-child app interactions have similarly found that children tend to lead app interactions, whereas parents tend to play the roles of "helper and commentator" (Griffith & Arnold, 2018). These parent-child role dynamics further contribute to the existing literature on joint media engagement (see e.g., Joan Ganz Cooney Center, 2014; Takeuchi & Stevens, 2011), which is driven by children's engagement in the activity itself (i.e., planning a ScratchJr project) and parents' interest in engaging the child (i.e., encouraging and coaching their child).

Although there were distinct quantitative differences in the observer, coach, and planner roles between parents and children, we found overall that parent-child dyads assumed and developed multiple roles over the course of their joint coding experience. In the case studies, for example, children taught their parents how to use various ScratchJr features, such as editing characters with the paint editor or changing the number of steps a character would move. Parents became involved in the planning process by proposing ideas for characters and blocks to add to their programs. The case studies also revealed roles outside of the five explored in ScratchJr Family Day surveys. For example, parents would sometimes act as "novices" by telling their child that they were not as knowledgeable, which would then empower the child to assume a teaching role. Parents would also act as "assistants" by holding the tablet for their child.

Our second research question involved exploring the relationships among children's coding interest (prior to attending ScratchJr Family Day), engagement in the five roles (planner, observer, teacher, coach, and playmate), and engagement in the three CT skills (algorithms, design process, and debugging). Of note, children's interest in coding was positively associated with their reported engagement in algorithmic thinking, the iterative design process, and debugging. In order to understand how children learn with ScratchJr, it is important to know which roles relate to domains of CT. The roles of teacher, coach, playmate, and planner were positively associated with parents' perception of CT skills while the observer role was not.

These associations suggest that when children are actively programing with the ScratchJr app, they are able to express their knowledge through teaching, coaching, and/or playing with their parent. These behaviors are viewed as helping to foster children's engagement in computational thinking. In addition, these findings promote ScratchJr as a playful, developmentally appropriate platform for young children to learn coding and engage in foundational CT skills. These results have implications for curriculum development and support the constructionist theory of learning that children not only learn by doing, but also by physically creating technological artifacts that they can share with others (Ackermann, 2001). Through the case studies, we saw examples of how children and parents would share ideas with one another or offer suggestions after testing part of their program. These collaborative behaviors enabled the child to engage more deeply with the tool and to learn from and alongside the parent, similarly to how they might engage with other digital apps such as e-books or tablet-based games.

Limitations and future work

There are several limitations of the two studies presented here. Our primary data source for the ScratchJr Family Day study consisted of pre- and post-surveys, which were all self-reported by participating parents and thus must be interpreted with some caution.

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). This limitation could have been resolved by videotaping events, conducting semi-structured interviews with parents and children, or surveying the children themselves, and future work should seek to employ these techniques. However, these methods seemed to be too intrusive for informal family workshop-style events and were beyond the scope of this particular project. Furthermore, the focus of the first study was to better understand parent perceptions of family-oriented coding in informal settings and particularly for early childhood, so the findings presented here still provide unique insight into those perspectives. 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 of the case studies.

Another study limitation of Study 1 was not having enough information about facilitator-hosted events. As the event was intended to be informal and adaptable to communities' needs, facilitators had the freedom to choose the number of co-facilitators at the event, monitor the time spent for each activity in the protocol, or adapt the activity prompts provided to families. These factors may have influenced how children and families engaged during the event, and as a result, how parents perceived their experiences overall.

Another limitation is that families self-selected to participate in both studies. 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 outcomes. It is also important to note that the majority of parents who attended ScratchJr Family Day events identified as female and all three parents from the Parent-Child Interactions study were mothers, which is not surprising because studies show that mothers often spend more time with their young children than do fathers (Connell, Lauricella, & Wartella, 2015; Lamb, 2000). However, fathers are typically more likely to engage in play and project-based activities with their children, particularly those that involve technology, programing, or other STEM-related activities (Bers, 2007). Having greater representation of mothers at informal family-based coding events like ScratchJr Family Days can perhaps help dismantle existing stereotypes surrounding STEM and computing fields (Metz, 2007; Steele, 1997). Future work should explore parent gender as a possible factor in impacting parent-child interactions with ScratchJr.

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. 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 engagement in coding and computational thinking. The findings here position the perspectives of the key stakeholders—parents—at the forefront of this movement.

Acknowledgments

The authors thank Kaitlyn Leidl for her initial contributions toward the project and members from the DevTech Research Group (namely Anna Kasagawa, Kathryn Tweel, Hyejin Im, and

Melissa Viezel, among others) for their assistance with recruitment and data collection. The authors also thank the families and facilitators who participated in this research.

Disclosure statement

The authors declare that there is no conflict of interest.

Funding

This project is supported by National Science Foundation grant DRL-1118664 and the Scratch Foundation.

ORCID

Madhu Govind http://orcid.org/0000-0002-6199-821X
Emily Relkin http://orcid.org/0000-0002-9031-6847
Marina Umaschi Bers http://orcid.org/0000-0003-0206-1846

References

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

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. doi: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. doi:10.1007/s10956-007-9074-2

Bers, M. U. (2018a). *Coding as a Playground: Programming and Computational Thinking in the Early Childhood Classroom*. New York, NY: Routledge press.

Bers, M. U. (2018b). Coding and Computational Thinking in Early Childhood: The Impact of ScratchJr in Europe. *European Journal of STEM Education*, *3*(3), 1–13. doi:10.20897/ejsteme/3868

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*. San Francisco, CA: 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.

Clements, D. H., & Gullo, D. F. (1984). Effects of computer programming on young children's cognition. *Journal of Educational Psychology*, *76*(6), 1051–1058. doi:10.1037/0022-0663.76.6.1051

Code.org.(2019). *K-12 Computer Science Policy and Implementation in the States*. Retrieved from https://docs.google.com/document/d/15zBdBbXUA-yEzxEq0VeWAEb9nXuGjmNFWNrYp6UdM8U/edit.

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. doi:10.1080/17482798.2015.997440

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. doi:10.1111/j.1467-8535.2010.01158.x

Flannery, L. P., Kazakoff, E., Bontá, P., Silverman, B., Bers, M. U., & 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. New York, NY: Acm. doi:10.1145/2485760.2485785

Fraillon, J., Ainley, J., Schulz, W., Duckworth, D., & Friedman, T. (2018). International Computer and Information Literacy Study. *ICILS 2018: Technical Report.*

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*, *13*, 1–19. doi:10.1080/17482798.2018.1489866

Hughes, M., & Greenhough, P. (1995). Feedback, adult intervention, and peer collaboration in initial Logo learning. *Cognition and Instruction*, *13*(4), 525–539. doi:10.1207/s1532690xci1304_4

International Society for Technology in Education and the Computer Science Teachers Association. (2011). Operational definition of computational thinking for K-12 thinking. *International Society for Technology in Education (ISTE) and the Computer Science Teachers Association (CSTA).* Available at: http://www.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf.

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.

Lamb, M. E. (2000). The history of research on father involvement: An overview. *Marriage & Family Review*, *29*, 23–42. doi:10.1300/J002v29n02_03

Lave, J., & Wenger, E. (1991). *Situated learning: legitimate peripheral participation*. Cambridge : Cambridge University Press.

Lin, J. M., & Liu, S. F. (2012). An investigation into parent-child collaboration in learning computer programming. *Educational Technology and Society*, *15*(1), 162–173.

Metz, S. S. (2007). Attracting the engineering of 2020 today. In R. Burke, M. Mattis, & E. Elgar (Eds.), *Women and minorities in science, technology, engineering and mathematics: Upping the numbers*. pp. 184–209. Northampton, MA: Edward Elgar Publishing.

National Early Literacy Panel (2008). Developing Early Literacy: Report of the National Early Literacy Panel.

Nederhof, A. J. (1985). Methods of coping with social desirability bias: A review. *European Journal of Social Psychology*, *15*, 263–280.

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

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

Portelance, D. J., Strawhacker, A., & Bers, M. U. (2015). Constructing the ScratchJr programming language in the early childhood classroom. *International Journal of Technology and Design Education*, *26*, 1–16. doi:10.1007/s10798-015-9325-0

Pretz, K. (2014). *Computer science classes for kids becoming mandatory: Countries are requiring coding to be taught in primary schools*. The Institute: The IEEE News Source. Retrieved from http://theinstitute.ieee.org/career-andeducation/preuniversity-education/computer-science-classes-for-kids-becoming-mandatory.

Resnick, M., & Silverman, B. (2005). Some reflections on designing construction kits for kids. Proceedings of Interaction Design and Children, 117–122. doi:10.1145/1109540.1109556

Roque, R. (2016). Family Creative Learning. In Peppler, K., Kafai, Y., & Halverson, E. (Eds.) *Makeology in K-12, Higher, and Informal Education*. New York, NY: Routeledge.

Roque, R., Lin, K., & Liuzzi, R. (2015). 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. Proceedings of the 12th International Conference of the Learning Sciences. 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. doi:10.1080/10645570701585129

Steele, C. M. (1997). A threat in the air: How stereotypes shape intellectual identity and performance. *American Psychologist*, *52* (6), 613–629. doi:10.1037/0003-066X.52.6.613

Swartz, M. I., & Crowley, K. (2004). Parent Beliefs about Teaching and Learning in a Children's Museum. *Visitor Studies*, *7*(2), 1. 5–16. doi:10.1080/10645570701585129

Strawhacker, A., & Bers, M. U. (2019). What they learn when they learn coding: Investigating cognitive domains and computer programming knowledge in young children. *Educational Technology Research and Development*, *67*, 541–575. doi:10.1007/s11423-018-9622-x

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, NY.

The Toy Association (2017). *STEM/STEAM Formula for Success.* Retrieved from https://www.toyassociation.org/App_Themes/toyassociation_resp/downloads/research/whitepapers/stemsteam-formulaforsuccess-2019.pdf.

U.K. Department for Education.(2013). *The National Curriculum in England: Framework document*. London: The Stationery Office.

Wing, J. M. (2006). Computational Thinking. *CACM Viewpoint*, 33–35. Retrieved from http://www.cs.cmu.edu/afs/cs/usr/wing/www/publications/Wing06.pdf.

Yu, J., & Roque, R. (2018). A survey of computational toys and kits for young children. Proceedings of the 17th ACM Conference on Interaction Design and Children. Trondheim, Norway. doi:10.1145/3202185.3202738

About the authors

Madhu Govind is a PhD student at the DevTech Research Group in the Eliot-Pearson Department of Child Study and Human Development at Tufts University. She received her B.S. in Child Studies and Neuroscience from Vanderbilt University and her M.A. in Child Study and Human Development from Tufts University. Madhu's research interests include developmentally appropriate technologies that inspire young children to tinker and create, collaborative learning opportunities for children and families, and pedagogical strategies for computer science integration in early childhood classrooms. Address correspondence to Madhu Govind, Tufts University, 105 College Ave, Medford, MA, 02155 USA. E-mail: madhu.govind@tufts.edu.

Emily Relkin is a PhD student at the DevTech Research Group in the Eliot-Pearson Department of Child Study and Human Development at Tufts University. Emily has her B.A. from Muhlenberg College in Psychology and her M.A. from Tufts University in Child Study and Human Development. Emily's research focuses on understanding and assessing the development of Computational Thinking abilities in young children.

Marina Umaschi Bers, PhD is a Professor and Chair of the Eliot-Pearson Department of Child Study and Human Development and a Professor at the Computer Science Department at Tufts University. She heads the interdisciplinary DevTech Research group, which focuses on designing studying innovative learning technologies to promote positive youth development. Dr. Bers received prestigious awards and has written three books: *Coding as a Playground: Programing and Computational Thinking in the Early Childhood Classroom* (2018; Routledge Press), *Designing Digital Experiences for Positive Youth Development: From Playpen to Playground* (2012; Oxford University Press) and *Blocks to Robots: Learning with Technology in the Early Childhood Classroom* (2008; Teacher's College Press). Dr. Bers has an MEd from Boston University and an MS and PhD from the MIT Media Lab. More on Dr. Bers: http://sites.tufts.edu/mbers01/.