Education Tech Research Dev (2021) 69:1579–1598
https://doi.org/10.1007/s11423-021-10011-w

RESEARCH ARTICLE

Taking coding home: analysis of ScratchJr usage in home and school settings

Apittha Unahalekhaka[1] · Marina Umaschi Bers[1]

Accepted: 21 May 2021 / Published online: 28 May 2021

Abstract
With a growing number of ScratchJr usage, over 19 million users worldwide, we examined the use in the United States of the free ScratchJr programming language, explicitly designed for young children ages 5–7, to learn how to code. Our objective was to explore children's usage of the ScratchJr tablet app at home and school settings. We analyzed usage data from Google Analytics in 1.5 years, comparing Scratchsr usage in the two different settings. Our dataset comprised a total of 4,352,802 coding sessions, generated by a daily average of 2525 home users and 9969 school users. The results suggested that, although children in both settings on average spent an equal duration with ScratchJr, children in home settings spent more time exploring advanced coding blocks and the paint editor compared to children at school. Further, children at school tended to use similar types of coding blocks across several days. In contrast, children at home were more likely to use a diversity of block categories and difficulty levels. The implications of this research are, first, that usage patterns may help us understand how children across settings learn to program differently. Second, based on these findings, it may be essential for parents at home and educators at school to consider using different approaches and strategies.

Keywords Early childhood · Computer science · Programming app · Educational settings · Quantitative analysis · Learning analytics

Introduction

ScratchJr is a popular introductory block-based programming language for young children ages 5–7 (Bers, 2018a, b). ScratchJr provides tools for children to create animated games and stories by putting together a sequence of graphical coding blocks (Fig. 1). In addition to programming blocks, ScratchJr has a painting tool for children to create and customize their characters and settings or to edit their photos onto the characters (Fig. 2). Since 2014,

✉ Apittha Unahalekhaka
apittha.u@tufts.edu

Marina Umaschi Bers
marina.bers@tufts.edu

1 Eliot-Pearson Department of Child Study and Human Development, Tufts University, Medford, MA 02115, USA

Fig. 1 Example of ScratchJr programming screen

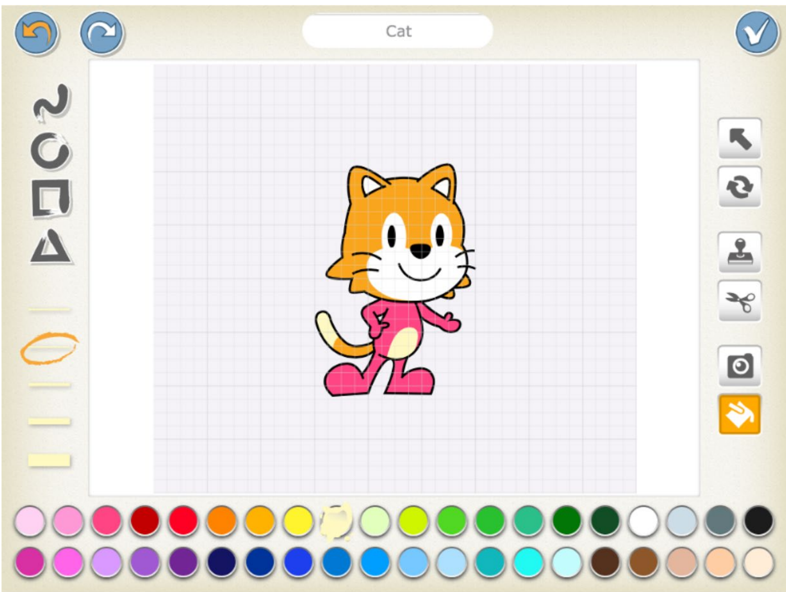

Fig. 2 Example of ScratchJr painting tool

ScratchJr has been available as a free app for iPad and Android tablets through the Apple Store or the Play Store. As of October 2020, ScratchJr had more than 19 million users, with over 59 million projects created and 84 million projects edited in 194 countries worldwide

(ScratchJr–DevTech Research Group, 2020). The United States had the greatest number of ScratchJr users, comprising 28% of the total downloads, or approximately 6 million users.

ScratchJr has an explicit educational design goal for children to learn coding across all settings: formal and informal, home and school. The app makes it easy to start coding while gradually learning complex computational concepts and skills. Furthermore, the app was designed as a "coding playground" (Bers, 2018a, b) so children can have the autonomy to creatively produce their own personally meaningful projects (Flannery et al., 2013). Given ScratchJr's features that allow children to code at home and school, it is essential to explore how children are using ScratchJr differently across settings.

Previous research studies have shown that using learning analytics, or data generated from students' learning tool usage, can help enhance students' performances by discovering their understanding, engagement, and learning patterns (Ifenthaler & Yau, 2020; Khalil & Ebner, 2015). Our research question was, are there any differences in ScratchJr usage patterns between home and school settings? To answer this research question, we analyzed the usage data in the United States collected through Google Analytics over a period of 1.5 years with average home users and school users per day of 2525 and 9969, respectively, and a total of 4,352,802 sessions. We analyzed four variables that have been previously shown to be related to computer science learning outcomes (Emerson et al., 2020; Price & Price-Mohr, 2018): time spent on coding compared to painting; complexity of coding blocks usage; categories of coding blocks usage; and consistency in the complexity of coding blocks usage.

This study's findings can contribute to the broader research on technological learning tools in early childhood by addressing the need to better understand similarities and differences in how young children used coding applications in formal and informal environments (Govind et al., 2020).

Background

Educational technology for home and school

Although digital technology for home and school usage has grown in the past decades, the trend has been more prominent at home, which might be due to increasing access to technology (Fraillon et al., 2014). For example, a global survey conducted in 2012 found that 93% of middle school students across 34 countries had computer devices at home for their school assignments (OECD, 2015). Another study by Rideout (2014) showed the responses from 1577 parents with children aged 2–10 in the United States that 34% of their children used educational media daily at home, and 80% of their children used educational media at least once per week. Among all mobile technologies, tablets have seen the fastest usage growth rate and are the most popular due to their affordability, accessibility, and ease of use (Oliemat et al., 2018; Reychav & Wu, 2015; Rideout, 2014). Although children use tablets in various settings, there has been a higher usage in informal settings than formal settings (Chee et al., 2017; Price et al., 2015).

This study set out to find if the trend observed in previous research—that *how* technology is being used across settings is as important—is also observable in ScratchJr usage. ScratchJr is designed for children ages 5–7 years old, which is an age range that commonly uses tablets for drawing and emergent writing (Couse & Chen, 2010). Some research found a positive correlation between emergent literacy skills and tablet reading-writing at home (Neumann, 2016). Researchers also found that tablets were able to facilitate personalized learning for children and increased self-expression and collaboration among peers (Algoufi, 2016; Bers, 2020; Couse & Chen, 2010; Wong, 2012).

Early childhood coding at school

While there has previously been low technology integration in early childhood classrooms, tablet usage among young learners at school increased almost two-fold from 2012 to 2014 in the United States (Blackwell et al., 2015). In early childhood classrooms, the use of tablets varied according to the teacher's different pedagogical approaches, professional development experiences, institutional support, and technology access (Radich, 2013; Young, 2016). A study conducted across preschools in Sweden revealed that technology and sciences were among the most commonly taught subjects with tablets (Otterborn et al., 2019). Furthermore, one of the objectives of using tablets was to teach how to "explore technology by creating and constructing," which can be done through coding or programming.

Numerous countries worldwide have agreed that computer science education should start from an early age (Noh & Lee, 2020; Webb et al., 2017). In the United States, due to the growing demand for computer science skills, the Computer Science for All initiative was launched in 2016 to provide computer science education for K-12 students across the country (FACT SHEET: President Obama Announces Computer Science For All Initiative, 2016). At this paper's writing, 34 states in the United States have integrated computer science in their curriculum from elementary-school to high-school levels (Computer Science Teachers Association, 2017; CS Advocacy, 2020).

Block-based programming languages such as Scratch provide an alternative to text-based programming languages and help students visualize computational concepts through color-coding blocks (Maloney et al., 2010). However, it is not enough in the early grades to copy models and pedagogies developed for middle school and high school even if the programming language is block-based (Grover et al., 2016; Strawhacker et al., 2018; Weintrop & Wilensky, 2017). A study by Strawhacker and Bers (2019) suggested a programming developmental progression for children from Kindergarten through second grade. They found that the children from the three grade levels could master different computational concepts. Therefore, developmentally appropriate programming languages and pedagogical approaches need to be put in place. An example of a developmentally appropriate coding learning tool is ScratchJr, a version of Scratch specifically designed for younger children (Bers, 2018a, b; Flannery et al., 2013).

In addition to the technology used, pedagogy is crucial in the classroom. Strawhacker et al. (2018) found that educators' teaching styles may impact students' programming learning. They conducted a study using ScratchJr with children in Kindergarten, first, and second grade and found that students had higher learning gains in programming when educators used certain teaching styles. These teaching styles include using flexible lesson plans, open-ended creative projects, and being responsive to students' needs. According to various studies conducted in schools with ScratchJr, young children could acquire coding concepts and skills when exposed to a well-designed curriculum. Examples of well-designed curriculums include step-by-step instructions on how each programming command works and engage children in creating open-ended final projects (Chou, 2019; Papadakis et al., 2016; Strawhacker et al., 2018). However, what happens when children are at home? Can children also learn programming without a fixed curriculum? How are the processes different or similar?

Early childhood coding at home

Various findings showed that high-quality home learning could benefit children's general development (Anders et al., 2012). Further, smartphone apps could help parents enhance their children's home learning experience (Jelley et al., 2019). By 2016, more than 80,000 apps in the Apple Store claimed to be educational for children (Shing & Yuan, 2016). Furthermore, a survey with over a thousand parents in the United States reported that 35% of children aged 2–10 years old play with educational apps on a mobile device weekly (Rideout, 2014). From these reports, parents seemed to increasingly want to use technological tools to immerse their young children in a richer home learning environment.

Despite the importance of home learning and mobile devices' popularity at home, there has been little research on how children learn programming at home. The number of existing studies that examined how children learn programming at school outnumbered the studies investigating how children learn at home. A handful of research studies found positive learning outcomes when parents collaboratively programmed with their children (Govind et al., 2020); however, the settings such as workshops, camps, and laboratories are not as informal as home (Hart, 2010; Lin & Liu, 2019; Roque et al., 2016; Sheehan et al., 2019).

Due to the growing educational app usage at home and the popularity of coding games, more research is needed to address how young children learn programming at home and school. Therefore, this study explored how children code with ScratchJr at home and school through their app usage.

Usage pattern as an indicator for learning

Usage data such as search volume, usage duration, and the number of hits can inform how children learn with educational, technological tools. Numerous studies have shown that students' usage data can predict learning successes (Ifenthaler & Yau, 2020). Zhang (2014) presented that internet search volume on a science simulation website (PhET) positively correlated with fourth-grade and eighth-grade students' academic performances. Another study found a positive relationship between the number of hits or views on students' portals and their final grades (Saqr et al., 2017). Duration of usage is another crucial factor for learning. Hu et al. (2014) reported that time-dependent variables such as the average time spent reviewing materials, the average time spent per session, and the total time spent online could predict undergraduate students' performance with online learning. Although researchers studied the connection between young children's usage patterns and multimodal learning analytics (e.g., eye-tracking, bodily movement, and electrodermal activity) to learning and developmental level, there has been an insufficient focus in the area of coding learning (Crescenzi-Lanna, 2020).

In computer science education, coding block usage can inform how well students learn and progress with the programming concepts (Emerson et al., 2020; Wang et al., 2017). The number of blocks and block types can particularly reflect differences in the quality of codes, effort, and understanding among novice programmers (Emerson et al., 2020). Furthermore, a study with primary students found that expert coders used less time and more complicated commands to make abstract programming concepts concrete compared to novices (Price & Price-Mohr, 2018). For early childhood, coding block complexity and types may be used to understand children's coding ability (Portelance et al., 2016). This study reported that second graders used more complex ScratchJr coding blocks than Kindergarteners and first graders.

Aligning with the previous studies on usage patterns, we analyzed children's ScratchJr usage data to answer our research question. We were interested in exploring the difference in the ScratchJr usage pattern between home and school, which may inform us how to design a better early childhood learning experience across settings.

Research methods

Research instrument

This study utilized Google Analytics to collect usage data from the ScratchJr app (Leidl et al., 2017). The usage data were tracked based on the user settings between home and school according to the users' responses when using the app for the first time. As Google Analytics did not collect any personally identifiable data (Clark et al., 2014; Google Analytics, 2020), no individual usage data were reported, and aggregated data based on home and school settings were used for all analyses in this paper.

Research participants

As mentioned above, no individual data from each user was reported due to how Google Analytics protects users' privacy while collecting data. Without any users' demographic data, we analyzed the fluctuating average usage values each day from home and school settings in the United States during the period between January 1, 2018, and June 30, 2019. Therefore, the average values of one day were equal to one data point in this study. During this study's period, there was an average of 2525 home users and 9969 school users per day. We assumed that users in this study were young children aged five and above, according to the age recommendation for ScratchJr in the Apple Store and the Play Store.

Research design

We analyzed ScratchJr usage data on Google Analytics in the United States between January 1, 2018, and June 30, 2019, during which home and school settings were included in Google Analytics data tracking. We examined home usage on the weekends and school usage on the weekdays, excluding the summer break for only the school usage group from July to August. As a result, the sample size for school was $n = 345$ days and for home was $n = 156$ days, with a total of 4,352,802 app usage sessions.

Before the primary analysis, we analyzed data from different days across home and school. A correlation test revealed that the numbers of coding blocks used at home and school were highly correlated with each other on weekdays ($r = .96$, $p < .01$) and were only mildly correlated on the weekends ($r = .42$, $p < .01$). In other words, on the weekdays, when the number of coding block usage increased at school, it was likely that the number of coding block usage will also increase at home, and vice versa. This trend was weaker on the weekends. A possible explanation for this finding was that children brought their tablets identified as "home" to use ScratchJr at school. In a recent survey of over 300,000 K-12 students and families, 62% of elementary school parents reported that they would purchase a personal tablet for their children to bring to school

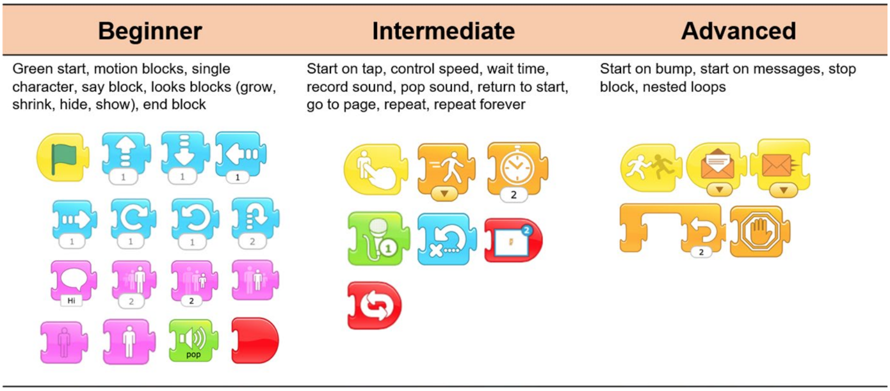

Fig. 3 ScratchJr's 28 coding blocks by levels

if permitted. Furthermore, 13% of students reported that they were allowed to bring their school-issued tablets home (Trends in Digital Learning: Students' Views on Innovative Classroom Models, 2014). Consequently, we decided to use home usage data on the weekends, when children would less likely go to school, and school usage data on the weekdays, excluding summer break. Although the lack of individual data was still a limitation to the study, we assumed that our home and school populations were independent.

Research measures

The two main functions that ScratchJr provides are coding and painting; however, in this study, we focused on the coding activity as it is the app's primary function. Studies by Emerson et al. (2020) and Price and Price-Mohr (2018) connected students' usage patterns such as time spent and coding block choices to learning outcomes; therefore, we based our measures on these findings.

From the entire dataset that Google Analytics provided, the variables that we analyzed were the number of coding blocks used, the number of users, the average session duration, and the number of sessions that the paint tool was open. Google Analytics recorded the number of coding blocks each time the user dragged a coding block out of the coding palette, as shown in Fig. 1. We also calculated the average session duration from the total duration of sessions divided by the number of sessions.

Block complexity

ScratchJr has 28 coding blocks in total. We split them into three coding block levels (beginner, intermediate, advanced) according to the block level categorization guidelines (Fig. 3) adapted from Portelance et al. (2016). The beginner blocks consist primarily of motion blocks such as forward, backward, and up, while the advanced blocks consist of more complex blocks such as repeat and looping commands. We used this measure to compare how much children in the home and school settings used each block-level daily.

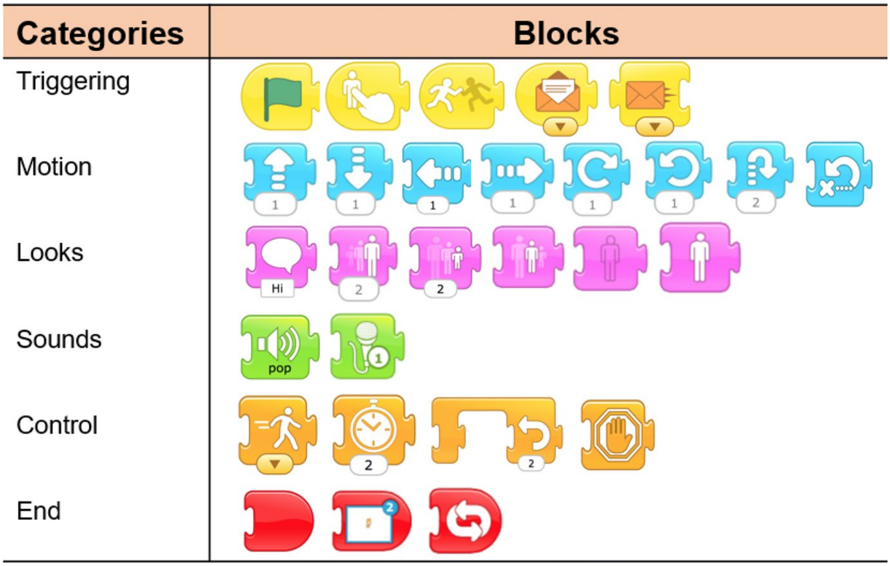

Fig. 4 ScratchJr's 28 coding blocks by categories

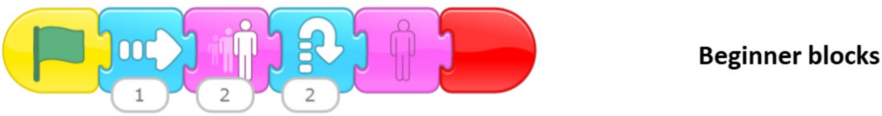

Fig. 5 An example of a sequence that only consists of beginner level blocks

Fig. 6 An example of a sequence that contains intermediate and beginner level blocks; the yellow, green, and orange blocks in this sequence are in intermediate level (Color figure online)

Block categories

Six coding block categories depending on the purpose of each coding block are shown in Fig. 4. This categorization came from the ScratchJr website on block descriptions (ScratchJr—Learn, 2020). In contrast to the block levels, the block categories represent each block's function, such as start the command or move the character in a particular direction. We used this measure to compare how much children in the home and school settings used each block category daily.

A comparison between three coding sequences with different block complexity levels is shown through Figs. 5, 6, 7. It is important to note that although motion blocks (blue) are at the beginner level, they are still fundamentally needed to complete a sequence of

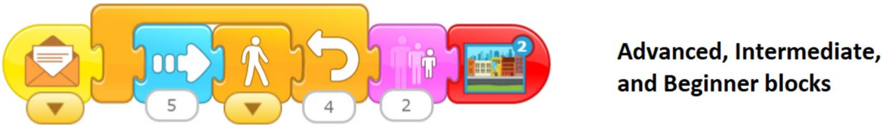

Fig. 7 An example of a sequence that contains all three block levels; the yellow and orange blocks in this sequence are in advanced level (Color figure online)

any level. Therefore, the examples in Figs. 6, 7 contain blocks from different levels. In Fig. 5, the sequence commands the character to move forward once, grow twice, jump twice, and then disappear.

The sequence in Fig. 6 has a mixture of intermediate and beginner blocks. This sequence will only start when a user physically taps at the character. The program then plays a voice recording block (green sound block). The character will then move forward five steps, speed up (orange control block), and ultimately shrink.

Figure 7 has a sequence that combines all three block complexity levels. This sequence starts with one of the most challenging blocks, receive message block (yellow trigger block), which will only run after receiving a message from a different sequence. Next, in this case, an orange repeat block tells the character to repeat (control block) the actions of move forward five steps and slow down (orange control block) four times. The last block is in an intermediate level block telling the project to go to the next page, of which there can be a maximum of four in one ScratchJr project.

Block consistency

The goal of studying block consistency was to see how constantly children at home and school used coding blocks of similar or different complexity levels across different days. To assess block consistency, we calculated the difference between the proportion of beginner block usage and the summation of the proportion of intermediate and advanced block usage. It is important to note that this study looked at the aggregated usage level at home and school in the United States. However, we will use individual-level usage to explain block consistency across 3 days in Fig. 8. For example, the number of beginner blocks (B) used by Child A was consistent across days relative to the other block levels (I or A). The difference in the block level ratios (consistency) for Child A from Day 1–3 were 1.00, .70, and .70. For Child B, there were lower block levels consistency across days as the difference in block ratios from Day 1–3 were .20, −.20, and 1.00.

Block consistency pattern is crucial as it may reflect children's exploratory coding style. Brennan and Resnick (2012) compared coding block usage between two children with different coding experiences with Scratch (a coding app for children ages 8–16). They showed that a child with more coding experience experimented with most block types and used various block levels. In contrast, a child with lower experience used similar and limited coding blocks across projects.

Session durations

Google Analytics can report session duration data in various usage scenarios. For example, we can filter the sessions during which children also use the paint editor feature in addition to coding. This study looked at three-session duration measures to compare the number of

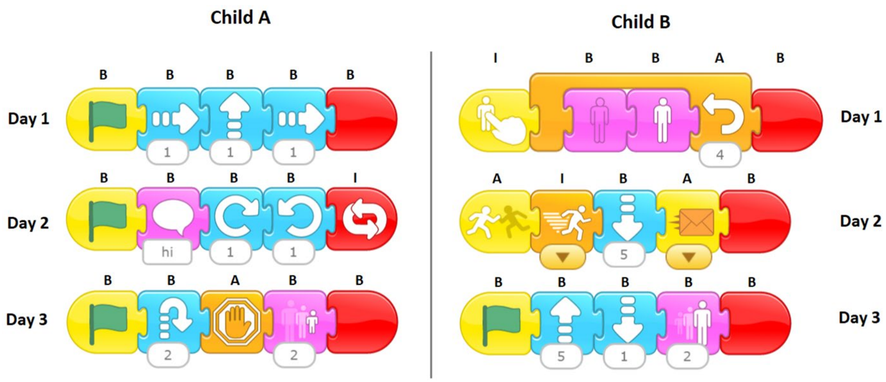

Fig. 8 A usage comparison across 3 days between child A who had higher consistency and mostly used beginner coding blocks, and child B who had lower consistency and a higher mixture of block levels. *B* Beginning block, *I* intermediate block, *A* advanced block

time children across settings spent coding (primary function) versus painting (secondary function). The first measure was the general session duration, including the average total session duration from all recorded sessions. In other words, children *may* have been using the paint editor feature in addition to coding in the general session duration. The second measure was the average total session duration for the sub-set of sessions in which children did *both* coding and painting. The third measure was the average total session duration in which the children *only* did coding without painting. Unfortunately, there was no average session duration for painting activity only as the primary function of ScratchJr is to code.

Data analysis

We used a statistical software package, SPSS version 24.0 (IBM Corp., 2016), to conduct multiple independent t tests. These t tests were used to compare the averaged usage behavior values between the two independent populations of children playing with ScratchJr at home and school. We also conducted z tests for the usage data reported in proportions, which revealed similar results to t tests; therefore, we decided only to report results from t tests.

We analyzed a total of 501 days with fluctuating numbers of users and coding blocks used each day. We checked all the appropriate inference assumptions for a t test analysis, including independence, normality, and homoscedasticity. We used Shapiro–Wilk to test for normality and F test to test for homogeneity of variance. All assumptions except normality were met with p values of above .05. Consequently, we addressed this violation in the descriptive analyses section, which was not a serious concern to us. We then conducted t tests to compare the difference between home and school settings. We examined the following variables—duration coding vs. painting, block complexity, block categories, and consistency in block complexity.

Duration of coding vs. painting activities

We analyzed the two main functions of ScratchJr: the use of programming blocks and the paint tool. To investigate the amount of programming and painting happening at home or school, we examined three average duration values from available data: (1) General sessions or the sessions with or without painting; (2) sessions with both coding and painting; (3) sessions with only coding, no painting.

Complexity

We compared the average proportion of the coding block levels used at home and school each day. We used proportion values for comparison due to the different number of users for each day and setting. To calculate the coding block-level (Fig. 3) proportions, we divided each block-level amount used by the total number of coding blocks used on that day. Consequently, there were three proportion values for beginner, intermediate, and advanced coding block levels.

Categories

Similarly, we examined the proportion of the coding block categories used on average across days and settings. We calculated the coding block category (Fig. 4) proportions by dividing the amount of each block category used by the total coding blocks used on that day. This resulted in six ratios for triggering, motion, looks, sounds, control, and end block categories.

Consistency

Lastly, we explored the consistency usage pattern from block complexity as it may show us how children across settings explored coding blocks. Specifically, we examined the variation of how children used coding block levels each day.

To compare the consistency usage, we found the difference between the proportion of beginner block level and the summation of the proportion of intermediate and advanced block levels. There were two main reasons why we compared the proportion of the beginner block-level against the combined intermediate and advanced coding block levels. First, beginner blocks are the fundamental level for the children to develop programming logic, whereas the higher block levels have additional purposes to make programming more complex. Secondly, a one-way ANOVA test revealed that children used beginner coding blocks four to seven times more than the intermediate and advanced coding blocks. Therefore, we decided to combine the two less commonly used block levels together.

Results

Descriptive analyses

After checking the descriptive statistics of all continuous variables in Table 1, 2, 3 the usage pattern at home across most variables were normally distributed with skewness and kurtosis values close to 0. The distributions of school setting variables were normally distributed by checking the histograms. However, some variables of school setting had high kurtosis values, showing a long thin tail in the distribution. This was not a highly concerning problem due to the large sample size in this study, which made results from *t* test analysis robust to the violation of the assumption of normality.

Table 1 Results of *t* tests and descriptive statistics ratio of session duration by settings in the US

Outcome	Settings		Mean difference	*t*	*df*	*Cohen's d*
	Home ($n = 156$)	School ($n = 345$)				
	M (SD)	*M (SD)*				
Avg. overall session duration	22.70 (2.33)	20.47 (2.48)	1.55	6.72*	317.45	.64
Avg. session duration with paint	24.51 (2.53)	21.43 (2.45)	3.08	12.74*	290.68	1.24
Avg. session duration without paint	12.99 (4.54)	17.46 (3.71)	−4.47	−10.78*	252.58	−1.08

All tests were significant at $*p < .001$; these three duration values were adjusted by Levene's test for equality of variances

Table 2 Results of *t* tests and descriptive statistics ratio of block levels by settings in the US

Outcome	Settings				Mean difference	*t*	*df*	*Cohen's d*
	Home		School					
	M (SD)	*n*	*M (SD)*	*n*				
Beginner	.73 (0.03)	156	.79 (0.03)	345	-.07	−22.78*	270.39	-2.23
Intermediate	.20 (0.03)	156	.15 (0.03)	345	.06	21.25*	289.29	2.13
Advanced	.07 (0.02)	156	.06 (0.01)	345	.01	7.69*	220.80	.93

All tests were statistically significant at $*p<.001$, these three-level values were adjusted by Levene's test for equality of variances

Table 3 Results of *t* tests and descriptive statistics ratio of block categories by settings in the US

Outcome	Settings				Mean difference	*t*	*df*	*Cohen's d*
	Home		School					
	M (SD)	*n*	*M (SD)*	*n*				
Triggering	.16 (0.02)	156	.15 (0.02)	345	.02	10.00*	260.17	.99
Motion	.38 (0.04)	156	.46 (0.04)	345	−.08	−20.47*	293.74	−1.98
Looks	.21 (0.03)	156	.21 (0.03)	345	−.004	−1.29	264.91	−0.13
Sounds	.08 (0.02)	156	.05 (0.02)	345	.03	15.35*	235.03	1.56
Control	.07 (0.02)	156	.05 (0.02)	345	.02	11.10*	328.69	1.05
End	.09 (0.01)	156	.07 (0.01)	345	.02	10.68*	276.33	1.05

$*p<.001$, these values were adjusted by Levene's test for equality of variances

Duration coding vs. painting

This study found that the children at home ($M=22.70$ min., $SD=2.34$ min.) spent approximately the same amount of time in the general sessions (coding with or without painting) as the children at school ($M=21.15$, $SD=2.49$). However, the children at home spent 3.08 min longer than the children at school in both coding and painting sessions. Furthermore, in the sessions with only coding, the children at home spent 4.47 fewer minutes on the average session than the children at school. All the three differences were statistically significant at $p<.001$, with medium to large Cohen's d values shown in Table 1. It is important to note that the ratios of the number of sessions with both coding and painting to the overall sessions were similar at home ($M=.71$, $SD=.05$) and school ($M=.72$, $SD=.04$). Therefore, these findings may suggest that the children at home spent a longer time painting and a shorter time coding than the children at school.

Complexity

To examine complexity, we looked at the coding block levels. Table 2 displays the *t* test results for coding block levels across two settings. On average, children at school ($M=.79$, $SD=.03$) used 7% more beginner blocks compared to the children at home ($M=.73$,

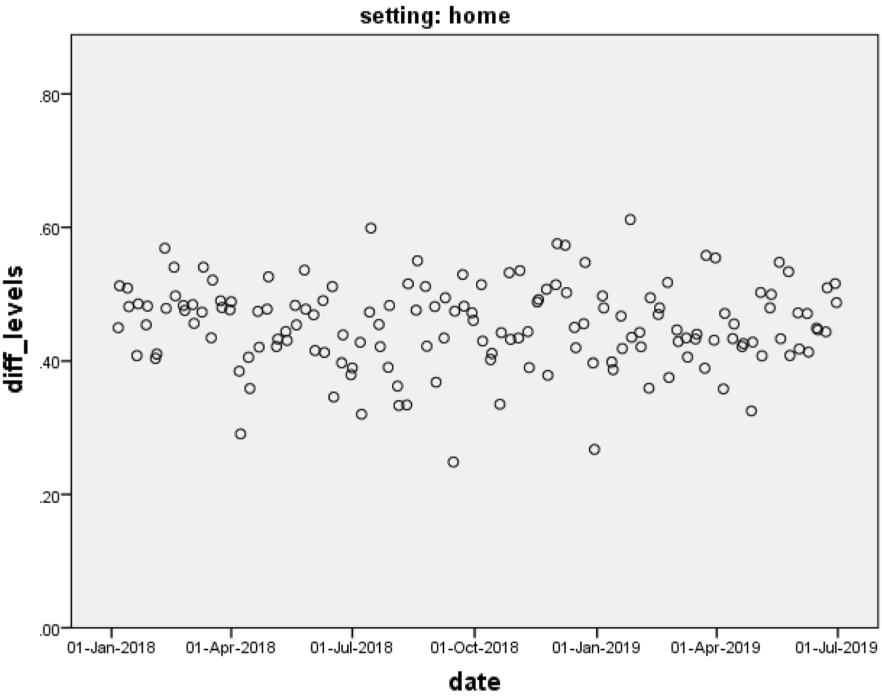

Fig. 9 Scatter plot of the difference in block levels used at home across days

SD = .03), this difference was statistically significant with a large effect size ($d = -2.23$). In contrast, children at home used 6% more intermediate blocks and 1% more advanced blocks than the children at school. These differences were statistically significant at large Cohen's d values of 2.13 and .93, respectively.

Categories

Table 3 contains the *t* test results for block categories across two settings. The mean differences in the proportion of all block categories across settings significantly differed, except for the difference in the looks block category. While the mean proportion difference of the other block categories ranged from 2% to 3%, the motion block category had the largest mean difference of 8% between home and school. Children at school ($M = .46$, $SD = .04$) used 8% more motion blocks than children at home ($M = .38$, $SD = .04$), $d = -1.98$.

Consistency

The scatter plots in Fig. 9, 10 present the difference in the beginner blocks' usage proportion from the two other block levels. The points on the home plot were more scattered than the school plot. Similarly, Levene's Test of Equality of Variances showed that the variances of the difference between levels were statistically different across settings ($F = 20.33$, $p < .001$). In other words, the children at home used different block levels less consistently across days than the children at school.

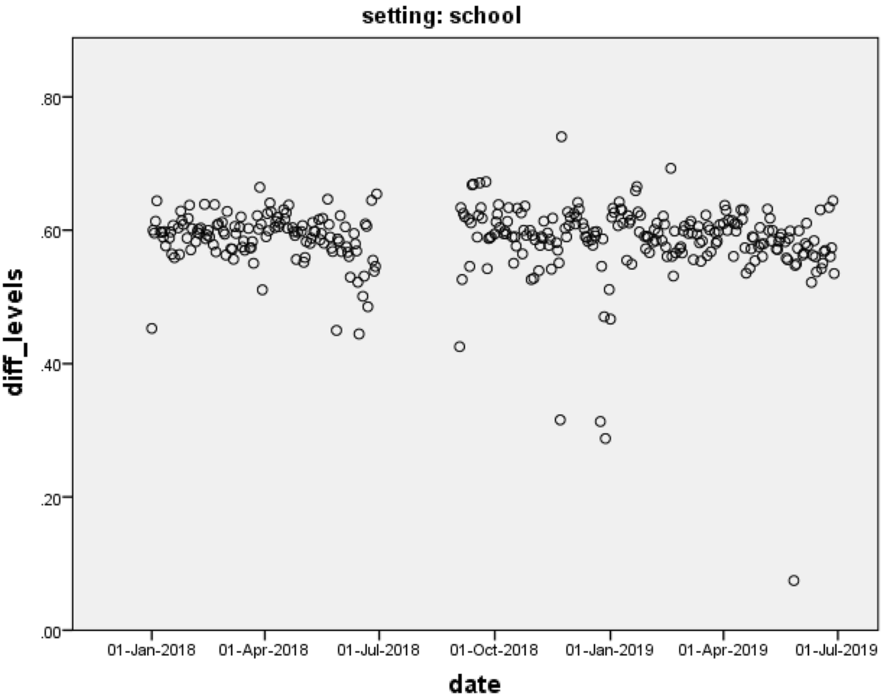

Fig. 10 Scatter plot of the difference in block levels used at school across days

Discussion and conclusion

With growing interest from schools and parents to start teaching computer science in early childhood, it is crucial to understand how tablet-based programming languages such as ScratchJr are being used both at home and school. Previous studies have found a connection between students' usage data and their learning performances (Emerson et al., 2020; Ifenthaler & Yau, 2020; Price & Price-Mohr, 2018). Therefore, we investigated ScratchJr usage data from Google Analytics by focusing on four usage areas—session duration, coding block complexity, coding block categories, and coding block consistency. This methodology was set to address our research goal of whether there were differences in how children across settings used ScratchJr in four usage areas. [1]

This study showed that children at home and school spent a similar amount of time using ScratchJr per session, approximately 20 min. However, our results suggested that children at home more often used ScratchJr for a different function than programming, such as painting characters, compared to children at school. This is credible, given that children at home were usually not following a pre-set curriculum but instead engaging in open-ended exploration. Although the children at home spent a shorter period on programming, they used more complex coding blocks. This might be because parents or older siblings can support children at home when creating their projects. Another possible explanation is that

[1] *Note.* July–August 2018 data got dropped from school only because it was during the summer break.

children who learn coding at school may choose to continue coding at home and experiment with more complex concepts in their free time. However, it is essential to note that we cannot conclude the quality of coding sequences created by just looking at the number of blocks used. Complex coding blocks used at home may not necessarily lead to functional sequences or creative projects. Our results also reported that the children at school focused more on motion blocks, one of the block categories, than the children at home. This finding aligned with previous research that studied the usage of ScratchJr with children from Kindergarten to second grade (Portelance et al., 2016). They found that motion blocks were most frequently used across all grades, which may be because they are the most foundational programming block level in ScratchJr to create stories and animations (Figs. 3, 4).

Furthermore, our study found a higher consistency of how children at school used the coding block levels each day, whereas the children at home appeared to use coding blocks more diversely. Although this study cannot identify children's learning strategy based on usage patterns, the higher consistency in block-level usage may suggest that children at school learn with ScratchJr by following guided instructions or a curriculum. In contrast, children at home may have a more open-ended and exploratory learning strategy.

We concluded that there were differences in how children used ScratchJr in various settings based on our data. Our results suggested that children using ScratchJr at school tend to use more foundational beginner level blocks, such as motion blocks. In contrast, intermediate and advanced coding blocks tend to be more popular among children at home. A possible explanation may be that children at school followed step-by-step instructions in class that focus on building mastery with the foundational ScratchJr blocks.

Additionally, children at home spent more time using the paint editor feature of the app. These findings suggested that the home setting may provide more freedom for children to engage in exploratory play with ScratchJr. This aligned with the previous research (Strawhacker et al., 2018) that showed how educators in formal learning settings hesitated to allow children to use painting tools or explore coding blocks on their own. From the same study, teachers revealed that children could focus and participate more if they could explore ScratchJr freely at the beginning of each class. Similarly, our findings suggested that children may be more motivated to explore creative and unknown elements of ScratchJr in their free-play time.

In this study, we concluded that children at home seemed to have more unstructured play and exploration with the app than children at school. However, we cannot claim that unstructured play at home leads to more meaningful learning as this study did not directly assess children's learning. Future studies will need to investigate this. As in many other development areas (DeCaro & Rittle-Johnson, 2012; Pramling Samuelsson & Johansson, 2006), child-directed free play with ScratchJr is an integral part of children's learning about programming. However, guided or step-by-step instruction still seems to be necessary, especially in early childhood education. A research study by Kirschner et al. (2006) explained that guided instruction is effective, especially when students have no prior knowledge in the learning area, and would be beneficial for students before tackling open-ended tasks.

This study supports that parents and educators may want to consider using different teaching approaches depending on the setting, especially when there is an increasing demand for remote learning going forward. Although this study did not capture the learning outcomes when children used ScratchJr differently across settings, that might be interesting to analyze in future work. Thus, more work is needed to understand the nature of various learning approaches and teaching strategies that will be most beneficial for young children across settings.

Limitations and future work

The main limitation of this study was the lack of individual and demographic data reported. With no basic demographic data such as age, we assumed that ScratchJr users in this study were children aged 5–7 years according to the app stores' age suggestion. Accordingly, we could not determine whether there were confounding factors related to users' demographic information that impacted the coding block usage pattern. Further, it was not possible to determine whether user groups from the two settings (home and school) were independent. We addressed this limitation by using data from certain days for analyses; we used weekdays for school setting data and weekend days for home setting data. Although this was still not sufficient to entirely ensure the two groups' independence, this approach allowed reasonable confidence that most weekday users would be at school or another formal learning setting, and most weekend users would be at home or another informal play setting. Future work should investigate children's ScratchJr use in the home and school settings by directly collecting data from identifiable participants.

Due to the limitations of the analytic data collected through the ScratchJr interface, there was also a lack of duration data on the sessions in which children only painted; without this data, we could not make a concrete claim about the difference between coding and painting activities across settings. Furthermore, Google Analytics tallied coding block usage as soon as users dragged the blocks out of the coding palette. We could not tell from the number of blocks alone whether the use was meaningful or purposeful. This paper's data suggested that children in home settings spent more time exploring advanced coding blocks and the paint editor than children at school. Although this could be interpreted to mean children at home were engaging in creative and self-expressive coding, it could also indicate that children at home played with novel blocks. If they could not understand them, they exited the coding environment to focus on the less complex activity of painting. At the same time, it might indicate that children at home received more support from adults or older children who guided them with more complex programming blocks. More research is needed to understand if the app usage reflects different children's motivations, as well as the context of how ScratchJr is used at home and schools.

Acknowledgements This work was generously funded by the National Science Foundation Grant No. DRL-1118664 and the Scratch Foundation. We would like to give special appreciation to Amanda Strawhacker and Tess Levinson for reviewing this manuscript. Lastly, we would like to thank everyone at the DevTech Research Group at Tufts University for contributing their ideas into this study.

Funding This study was funded by NSF (DRL-1118664) and the Scratch Foundation.

Data availability The usage data collected by Google Analytics is non-personally identifiable data as also clarified to the users on the ScratchJr privacy website, https://www.scratchjr.org/privacy. There is no individual data reported and this study analyzed data from the entire US level.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical Approval Not applicable.

Informed consent Not applicable.

References

Algoufi, R. (2016). Using tablet on education. *World Journal of Education, 6*(3), 113. https://doi.org/10.5430/wje.v6n3p113

Anders, Y., Rossbach, H.-G., Weinert, S., Ebert, S., Kuger, S., Lehrl, S., & von Maurice, J. (2012). Home and preschool learning environments and their relations to the development of early numeracy skills. *Early Childhood Research Quarterly, 27*(2), 231–244. https://doi.org/10.1016/j.ecresq.2011.08.003

Bers, M. U. (2018). Coding, playgrounds and literacy in early childhood education: The development of KIBO robotics and ScratchJr. *IEEE Global Engineering Education Conference (EDUCON), 2018*, 2094–2102. https://doi.org/10.1109/EDUCON.2018.8363498

Bers, M. U. (2018). Coding and computational thinking in early childhood: The impact of ScratchJr in Europe. *European Journal of STEM Education*. https://doi.org/10.20897/ejsteme/3868

Bers, M. U. (2020). *Coding as a playground: Programming and computational thinking in the early childhood classroom*. Routledge.

Blackwell, C. K., Wartella, E., Lauricella, A. R., & Robb, M. B. (2015). Technology in the lives of educators and early childhood programs: Trends in access, use, and professional development from 2012 to 2014. Retrieved November 20, 2019, from http://www.fredrogerscenter.org/wpcontent/uploads/2015/07/Blackwell-Wartella-Lauricella-Robb-Tech-in-the-Lives-of-Educatorsand-Early-Childhood-Programs.pdf.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In: *Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada* (Vol. 1, p. 25).

Chee, K. N., Yahaya, N., Ibrahim, N. H., & Hasan, M. N. (2017). Review of mobile learning trends 2010–2015: A meta-analysis. *Journal of Educational Technology & Society, 20*(2), 113–126.

Chou, P.-N. (2019). Using ScratchJr to foster young children's computational thinking competence: A case study in a third-grade computer class. *Journal of Educational Computing Research*. https://doi.org/10.1177/0735633119872908

Clark, D. J., Nicholas, D., & Jamali, H. R. (2014). Evaluating information seeking and use in the changing virtual world: The emerging role of google analytics. *Learned Publishing, 27*(3), 185–194. https://doi.org/10.1087/20140304

Computer Science Teachers Association (2017). K-12 computer science standards, Revised 2017 [PDF File]. Retrieved October 5, 2019, from https://www.doe.k12.de.us/cms/lib/DE01922744/Centricity/Domain/176/CSTA%20Computer%20Science%20Standards%20Revised%202017.pdf

Couse, L. J., & Chen, D. W. (2010). A tablet computer for young children? Exploring its viability for early childhood education. *Journal of Research on Technology in Education, 43*(1), 75–96. https://doi.org/10.1080/15391523.2010.10782562

Crescenzi-Lanna, L. (2020). Multimodal Learning Analytics research with young children: A systematic review. *British Journal of Educational Technology, 51*(5), 1485–1504. https://doi.org/10.1111/bjet.12959

CS Advocacy. (n.d.). Retrieved January 3, 2021, from https://advocacy.code.org/

DeCaro, M. S., & Rittle-Johnson, B. (2012). Exploring mathematics problems prepares children to learn from instruction. *Journal of Experimental Child Psychology, 113*(4), 552–568. https://doi.org/10.1016/j.jecp.2012.06.009

Emerson, A., Smith, A., Rodriguez, F. J., Wiebe, E. N., Mott, B. W., Boyer, K. E., & Lester, J. C. (2020). Cluster-based analysis of novice coding misconceptions in block-based programming. In: *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 825–831. https://doi.org/10.1145/3328778.3366924

FACT SHEET: President Obama Announces Computer Science For All Initiative. (2016). *Whitehouse.Gov*. Retrieved October 7, 2019, from https://obamawhitehouse.archives.gov/the-press-office/2016/01/30/fact-sheet-president-obama-announces-computer-science-all-initiative-0

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. In: *Proceedings of the 12th International Conference on Interaction Design and Children—IDC 13*, 1–10. https://doi.org/10.1145/2485760.2485785

Fraillon, J., Ainley, J., Schulz, W., Friedman, T., & Gebhardt, E. (2014). Students' use of and engagement with ICT at home and school. In J. Fraillon, J. Ainley, W. Schulz, T. Friedman, & E. Gebhardt (Eds.), *Preparing for life in a digital age* (pp. 125–166). Springer International Publishing. https://doi.org/10.1007/978-3-319-14222-7_6

Google Analytics. (2020). *Firebase*. Retrieved December 15, 2019, from https://firebase.google.com/docs/analytics

Govind, M., Relkin, E., & Bers, M. U. (2020). Engaging children and parents to code together using the ScratchJr app. *Visitor Studies, 23*(1), 46–65. https://doi.org/10.1080/10645578.2020.1732184
Grover, S., Pea, R., & Cooper, S. (2016). Factors influencing computer science learning in middle school. In: *Proceedings of the 47th ACM Technical Symposium on Computing Science Education—SIGCSE '16*, 552–557. https://doi.org/10.1145/2839509.2844564
Hart, M. L. (2010). Making contact with the forgotten k-12 influence: Are you smarter than your 5th grader? In: *Proceedings of the 41st ACM Technical Symposium on Computer Science Education—SIGCSE 10*, 254. https://doi.org/10.1145/1734263.1734349
Hu, Y.-H., Lo, C.-L., & Shih, S.-P. (2014). Developing early warning systems to predict students' online learning performance. *Computers in Human Behavior, 36*, 469–478. https://doi.org/10.1016/j.chb.2014.04.002
IBM Corp. Released 2016. IBM SPSS Statistics for Windows, Version 24.0. Armonk, NY: IBM Corp.
Ifenthaler, D., & Yau, J.Y.-K. (2020). Utilising learning analytics to support study success in higher education: A systematic review. *Educational Technology Research and Development, 68*(4), 1961–1990. https://doi.org/10.1007/s11423-020-09788-z
Jelley, F., Sylva, K., & Villalobos, V. O. (2019). Supporting children's learning at home through smartphone apps for parents. *The Routledge international handbook of learning with technology in early childhood* (pp. 324–337). Routledge. https://doi.org/10.4324/9781315143040-23
Khalil, M., & Ebner, M. (2015). A STEM MOOC for school children—what does learning analytics tell us? *International Conference on Interactive Collaborative Learning (ICL), 2015*, 1217–1221. https://doi.org/10.1109/ICL.2015.7318212
Kirschner, P. A., Sweller, J., & Clark, R. E. (2006). Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educational Psychologist, 41*(2), 75–86. https://doi.org/10.1207/s15326985ep4102_1
Leidl, K. D., Bers, M. U., & Mihm, C. (2017). Programming with ScratchJr: A review of the first year of user analytics. In: *Proceedings of International Conference on Computational Thinking Education* (pp. 116–121).
Lin, J.M.-C., & Liu, S.-F. (2019). An investigation into parent–child collaboration in learning computer programming. *Educational Technology & Society, 15*(1), 162–173.
Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The scratch programming language and environment. *ACM Transactions on Computing Education, 10*(4), 1–15. https://doi.org/10.1145/1868358.1868363
Neumann, M. M. (2016). Young children's use of touch screen tablets for writing and reading at home: Relationships with emergent literacy. *Computers & Education, 97*, 61–68. https://doi.org/10.1016/j.compedu.2016.02.013
Noh, J., & Lee, J. (2020). Effects of robotics programming on the computational thinking and creativity of elementary school students. *Educational Technology Research and Development, 68*(1), 463–484. https://doi.org/10.1007/s11423-019-09708-w
OECD. (2015). *Students, Computers and Learning: Making the Connection*, PISA, OECD Publishing, Paris. https://doi.org/10.1787/9789264239555-en.
Oliemat, E., Ihmeideh, F., & Alkhawaldeh, M. (2018). The use of touch-screen tablets in early childhood: Children's knowledge, skills, and attitudes towards tablet technology. *Children and Youth Services Review, 88*, 591–597. https://doi.org/10.1016/j.childyouth.2018.03.028
Otterborn, A., Schönborn, K., & Hultén, M. (2019). Surveying preschool teachers' use of digital tablets: General and technology education related findings. *International Journal of Technology and Design Education, 29*(4), 717–737. https://doi.org/10.1007/s10798-018-9469-9
Papadakis, S., Kalogiannakis, M., & Zaranis, N. (2016). Developing fundamental programming concepts and computational thinking with ScratchJr in preschool education: A case study. *International Journal of Mobile Learning and Organisation, 10*(3), 187. https://doi.org/10.1504/IJMLO.2016.077867
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
Pramling Samuelsson, I., & Johansson, E. (2006). Play and learning—inseparable dimensions in preschool practice. *Early Child Development and Care, 176*(1), 47–65. https://doi.org/10.1080/0300443042000302654
Price, C. B., & Price-Mohr, R. M. (2018). An evaluation of primary school children coding using a text-based language (Java). *Computers in the Schools, 35*(4), 284–301. https://doi.org/10.1080/07380569.2018.1531613
Price, S., Jewitt, C., & Crescenzi, L. (2015). The role of iPads in preschool children's mark making development. *Computers & Education, 87*, 131–141. https://doi.org/10.1016/j.compedu.2015.04.003

Radich, J. (2013). Technology and interactive media as tools in early childhood programs serving children from birth through age 8. *Every Child, 19*(4), 18–19. Retrieved January 20, 2020, from https://search.informit.com.au/documentSummary;dn=728719428486158;res=IELAPA

Reychav, I., & Wu, D. (2015). Mobile collaborative learning: The role of individual learning in groups through text and video content delivery in tablets. *Computers in Human Behavior, 50*, 520–534. https://doi.org/10.1016/j.chb.2015.04.019

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.

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.* Singapore.

Saqr, M., Fors, U., & Tedre, M. (2017). How learning analytics can early predict under-achieving students in a blended medical education course. *Medical Teacher, 39*(7), 757–767. https://doi.org/10.1080/0142159X.2017.1309376

ScratchJr – DevTech Research Group. (2020). Retrieved January 3, 2021, from https://sites.tufts.edu/devtech/research/scratchjr/

ScratchJr—Learn. (2020). Retrieved January 4, 2021, from https://scratchjr.org/

Sheehan, K. J., Pila, S., Lauricella, A. R., & Wartella, E. A. (2019). Parent–child interaction and children's learning from a coding application. *Computers & Education, 140*, 103601. https://doi.org/10.1016/j.compedu.2019.103601

Shing, S., & Yuan, B. (2016). Apps developed by academics. *Journal of Education and Practice, 7*, 1–9.

Strawhacker, A., Lee, M., & Bers, M. U. (2018). Teaching tools, teachers' rules: Exploring the impact of teaching styles on young children's programming knowledge in ScratchJr. *International Journal of Technology and Design Education, 28*(2), 347–376. https://doi.org/10.1007/s10798-017-9400-9

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*(3), 541–575. https://doi.org/10.1007/s11423-018-9622-x

Trends in digital learning: Students' views on innovative classroom models. (2014). *Project tomorrow*. Retrieved October 10, 2019, from https://tomorrow.org/speakup/2014_OnlineLearningReport.html

Wang, L., Sy, A., Liu, L., & Piech, C. (2017). Learning to Represent Student Knowledge on Programming Exercises Using Deep Learning. *International Educational Data Mining Society*.

Webb, M., Davis, N., Bell, T., Katz, Y. J., Reynolds, N., Chambers, D. P., & Sysło, M. M. (2017). Computer science in K-12 school curricula of the 21st century: Why, what and when? *Education and Information Technologies, 22*(2), 445–468. https://doi.org/10.1007/s10639-016-9493-x

Weintrop, D., & Wilensky, U. (2017). Comparing block-based and text-based programming in high school computer science classrooms. *ACM Transactions on Computing Education, 18*(1), 1–25. https://doi.org/10.1145/3089799

Wong, L.-H. (2012). A learner-centric view of mobile seamless learning. *British Journal of Educational Technology, 43*(1), E19–E23. https://doi.org/10.1111/j.1467-8535.2011.01245.x

Young, K. (2016). Teachers' attitudes to using iPads or tablet computers; implications for developing new skills pedagogies and school-provided support. *TechTrends, 60*(2), 183–189. https://doi.org/10.1007/s11528-016-0024-9

Zhang, M. (2014). Who are interested in online science simulations? Tracking a trend of digital divide in Internet use. *Computers & Education, 76*, 205–214. https://doi.org/10.1016/j.compedu.2014.04.001

Apittha Unahalekhaka is a doctoral student at the Eliot-Pearson Department of Child Study and Human Development at Tufts University, under the Developmental Technologies research group. Her research interests include using data science for education and studying the relationship between socio-emotional and cognitive development in early childhood through developmentally appropriate technologies.

Marina Umaschi Bers is a professor and chair at the Eliot-Pearson Department of Child Study and Human Development and an adjunct professor in the Computer Science Department at Tufts University. She heads the interdisciplinary Developmental Technologies research group which focuses on developing, implementing, and evaluating innovative learning technologies to promote positive technological development.