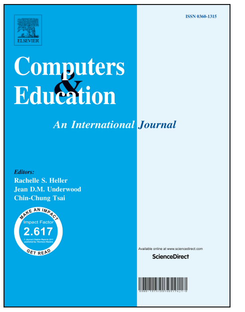

Learning to code via tablet applications: An evaluation of *Daisy the Dinosaur* and *Kodable* as learning tools for young children

Sarah Pila, Fashina Aladé, Kelly J. Sheehan, Alexis R. Lauricella, Ellen A. Wartella

PII: S0360-1315(18)30242-2

DOI: 10.1016/j.compedu.2018.09.006

Reference: CAE 3449

To appear in: *Computers & Education*

Received Date: 13 December 2017

Revised Date: 7 September 2018

Accepted Date: 15 September 2018

Learning to Code via Tablet Applications: An Evaluation of *Daisy the Dinosaur* and *Kodable* as Learning Tools for Young Children

Sarah Pila, Fashina Aladé, Kelly J. Sheehan, Alexis R. Lauricella, and Ellen A. Wartella

Northwestern University

Author Note

Sarah Pila, Media, Technology, and Society Program, Northwestern University; Fashina Aladé[1], Media, Technology, and Society Program, Northwestern University; Kelly J. Sheehan[2], Department of Communication Studies, Northwestern University; Alexis R. Lauricella[3], Department of Communication Studies, Northwestern University; Ellen A. Wartella, Department of Communication Studies, Northwestern University.

This research was supported by a grant from the National Science Foundation Reese Program # DRL-1252121. Thank you to Ann C. Gadzikowski, the teachers, students, and families involved in the Story Code Alpha classroom. The authors also thank Kinnari Atit, Jue Wue, Grace Oni**,** Zachary Lochmueller**,** Leanne Beaudoin-Ryan, David Uttal, and the members of the Center on Media and Human Development for their assistance with data collection, analysis, and feedback on earlier drafts of this manuscript.

Correspondence concerning this article should be addressed to Sarah Pila, MTS PhD Program, Northwestern University, 2240 Campus Drive, Room 2-101, Evanston, IL 60208-3545. Email: sarah.c.pila@u.northwestern.edu

[1] Present address at Communication Arts and Sciences, Michigan State University

[2] Present address at Age of Learning, Inc.

[3] Present address at Erikson Institute

Abstract

Despite the growing number of digital apps designed to teach coding skills to young children, we know little about their effectiveness. To formally explore this question, we conducted a naturalistic observation of a one-week program designed to teach foundational coding skills (i.e., sequencing, conditions, loops) to young children (N = 28, M_{age} = 5.15 years) using two tablet applications: *Daisy the Dinosaur* and *Kodable*. Pre- and post-assessments measured familiarity with technology, appeal of coding apps, knowledge of *Daisy* commands, ability to play *Kodable*, and conceptual understanding of coding. Participants improved in their knowledge of *Daisy* commands (i.e., *move*, *grow*, *jump*) and *Kodable* gameplay (i.e., placing arrows in the correct sequence to move a character through a maze), but did not improve in their ability to verbally explain what coding is. Appeal of the games was significantly related to children's learning of *Daisy* commands, but child gender was not related to either *Daisy* or *Kodable* learning outcomes. Results suggest that young children can learn foundational coding skills via apps, especially when the apps are appealing to children.

Keywords: Apps, Coding, computational thinking, Digital games, Educational Technology, STEM

Learning to Code Using Tablet Apps: An Evaluation of *Daisy the Dinosaur* and *Kodable* as Learning Tools for Young Children

1. Introduction

Despite the growing importance of science, technology, engineering, and math (STEM) jobs, globally, there stand to be more jobs in STEM than people to fill them. In the United States alone, employment for computer occupations is expected to increase 12.5 percent from 2014-2024, resulting in almost 500,000 new jobs (Fayer, Lacey, & Watson, 2017), but some believe that school systems are not producing enough graduates to fill this need (e.g. Kelly, 2012; Rothwell, 2014). To address this forthcoming gap, policy makers have pushed for greater emphasis on STEM initiatives in early education settings (U.S. Department of Education, 2015, 2016, n.d.). One of these initiatives is focused specifically on teaching concepts of computer science (e.g., coding/programming) in order to address both the “T” and “E” of STEM.

Some believe that early experiences with computer science concepts are vital and should be made widely available to young children (Bers, 2017; O'Dell, 2013). Research on coding and programming curricula in early childhood suggests that children as young as age five can learn foundational coding skills such as sequencing (i.e., entering commands in a particular order so that they will be carried out by the computer in the correct order) and conditions (i.e., “if-then” statements that tell the computer to perform different actions based on whether or not a certain criterion has been met) (Fessakis, Gouli, & Mavroudi, 2012; Karadeniz, Samur, & Ozden, 2014; Kazakoff & Bers, 2014). For these reasons, a number of digital applications (apps) that are designed to teach young children coding skills in a fun, game-like way have become popular in recent years (e.g. García-Peñalvo et al., 2016). However, there is little empirical research on the features of these apps and their effectiveness in teaching coding skills for preschoolers and early elementary children. The current study seeks to add to the literature on these programs by providing evidence from a summer camp classroom observation where children were taught to code via two tablet-based apps.

2. Literature Review

2.1. Learning Through/By Media

Nearly 50 years of research has shown that children can learn a host of academic skills from curriculum-based television (e.g. Anderson, Lavigne, & Hanson, 2013; Ball & Bogatz, 1970; Fisch, 2004b), but we are just starting to amass research on whether children are able to learn from interactive tablet applications that are designed to be educational (Hirsh-Pasek et al., 2015). Most of the research on children's learning from tablet applications has focused on literacy outcomes by comparing e-books to traditional print materials. Of these studies, conclusions are mixed. Krcmar and Cingel (2014) found that children comprehended a traditional storybook better than an electronic version of the same book, while Etta, Kirkorian, and Choi (2017) and Lauricella, Barr, & Calvert (2014) saw no difference in children's comprehension from the two different types of books. The research on children's learning of STEM topics from digital apps is even more limited, though it does appear that children can transfer learning from tablet games to particular STEM tasks. For example, Aladé, Lauricella, Beaudoin-Ryan, and Wartella (2016) found that children exposed to a measuring tablet game (either by playing the game interactively or watching a pre-recorded video of the game) did better on a measuring transfer task than the control group, who played a comparable non-measuring game. Other new research also echoes these findings, suggesting that exposure to high quality educational apps that are designed to teach STEM skills can support desirable STEM learning outcomes (Huber et al., 2016; Schroeder & Kirkorian, 2016).

2.2. Learning Coding Skills Through Games and Apps

Coding, the process of assigning a set of symbols that can be interpreted by a computer or piece of software, is considered a building block of technology competence, and has been defined as a new modern literacy (Prensky, 2008). The process of coding consists of several essential, foundational concepts: sequencing, conditions, and loops (repetition). Bers (2012) describes sequencing as the “sequence of instructions described in a program and acted out in order by a robot [or computer]. Each block has a specific meaning. The order of the blocks is important.” Conditions (or conditional statements) are “if, then, else” statements that tell the program which actions to perform. For example, “if path hits red, then up.” Loops are described as “a sequence of instructions that can be modified to occur over and over again. The sequence can be qualified with additional information. For example, loops can repeat forever or for a concrete number of times” (Bers, 2012).

It should be noted that ‘programming’ is another widely used term for a closely related set of skills and concepts. Unlike coding, which deals specifically with assigning meaning to symbols, the term ‘programming’ refers to a more advanced process of designing and developing instructions for executing a specific computing task (Morgado, Cruz, & Kahn, 2010). Because these two processes are associated with similar learning outcomes, this paper cites research that looks at both coding and programming. Though the focus of our study is coding, whenever possible, we use the original terminology provided by the authors of the empirical research that we cite.

Given the potential importance of foundational coding skills for STEM learning, and the fact that existing research suggests young children can learn various skills and concepts through playing apps and computer games, it is important to consider whether young children can learn foundational coding skills through games as well. Traditionally, research on coding and programming with young children has been conducted using two different media; computer programs and tangible robotics. The earliest studies of young children and coding have evaluated children's learning from the computer-based Logo programming language. In describing the pilot testing of Logo, Papert (1980) first suggested that the programming language could allow even preschoolers to grasp the fundamental skills of programming. Indeed, Clements and Gullo (1984) found that first graders who were taught the Logo programming language over the course of 12 weeks improved significantly in their ability to describe directions (a cognitive skill Logo emphasizes) compared to a control group. Similarly, Golden, Blackburn, Scarlato, Free, and Degelman (1986) found that kindergarteners who used Logo for five weeks performed significantly better on a rule-following task (an element of Logo language) compared to a control group that was not exposed to Logo.

Developmentally, however, coding from a computer with text languages might be difficult for younger children who are not yet reading and who may not have the motor skills to control a mouse. To address these concerns, researchers have developed tangible interface design robots for teaching coding in the early years. Research on coding and programming robots suggests that children as young as age four can learn to correctly sequence a robot to perform a specific task (e.g. Kazakoff & Bers, 2014; Kazakoff, Sullivan, & Bers, 2012; Sullivan, Kazakoff, & Bers, 2013). However, these studies often include only one outcome variable, lacking nuance in their findings. It is also possible that using these real-world manipulatives in a tangible interface may lack ecological validity for real-world coding, which is typically done on a computer without such manipulatives.

Tablet-based apps may be an ideal platform for young children to learn coding skills because apps are screen-based like coding on a computer, but also support non-text languages Further, apps are considered easy-to-use by young children (Siegle, 2013), and access to tablets is nearly ubiquitous for families in the U.S. today (Rideout, 2017).

Research on apps developed to teach coding to preschoolers is relatively new and has been largely focused around the development of *ScratchJr* (Flannery et al., 2013; Portelance, Strawhacker, & Bers, 2015; Strawhacker, Lee, Caine, & Bers, 2015). These researchers found that young children can engage in developmentally-appropriate computer programming through the game. *ScratchJr* is not the only popular app designed to teach these skills to this age group, however. Similar apps like *Kodable* and *Daisy the Dinosaur* have also become popular in recent years (Common Sense Media, n.d.), but research on these apps is limited. One study evaluating *Kodable* found that all but one of the 25 five-year-olds in their study were able to master the sequencing level of the game by the end of one week of play (Karadeniz et al., 2014), but it is unclear how exactly the researchers measured such mastery. The quantitative research done so far to evaluate young children's learning of coding/programming concepts via apps has laid an important foundation, but outcome measures are often not systematic and have been based primarily on research-as-design protocols.

2. 3. Moderating Factors

While research points to the possibility of children learning foundational coding and programming skills through media, we know that media do not affect users uniformly. That is, individual differences matter, and cognitive abilities as well as motivational differences can affect how well an individual child is able to comprehend and learn from media (Aladé & Nathanson, 2016; Piotrowski & Valkenburg, 2015; Valkenburg & Peter, 2013). Specifically, two possible moderating factors might impact children's learning of foundational coding skills: participant gender and participant interest in the apps used in the intervention.

2.3.1 Gender. Gender differences have played an important role in children's interest and achievement in STEM disciplines. There are disproportionately fewer women entering the science, technology, engineering, and math pipelines (Beede et al., 2011; Ceci, Williams, & Barnett, 2009; National Science Foundation, 2013) , and women choose STEM occupations less often than men do (Ceci et al., 2009; Wang, Eccles, & Kenny, 2013). In the computing world specifically, only 26% of professional computing occupations are held by women (National Center for Women & Information Technology, 2017). For women who do continue in STEM careers, there are a number of barriers towards successful integration and later achievement in the workplace (Ceci & Williams, 2011; Williams & Ceci, 2012). However, evidence on differences in STEM learning by gender is mixed. Some researchers have found that males perform better, on average, than females at spatial reasoning tasks (Linn & Petersen, 1985; Voyer, Voyer, & Bryden, 1995) and mental rotation (e.g. Moore & Johnson, 2008). In particular, Moore and Johnson (2008) used a mental rotation task with five-month-old infants and demonstrated an early gender difference in favor of males. Since such spatial ability is related to later involvement in STEM majors and occupations (Wai, Lubinski, & Benbow, 2009), these gender differences are important to note. Other studies, though, have found no differences or only very trivial differences in learning academic STEM content by gender (Petersen & Hyde, 2014; Voyer & Voyer, 2014). Therefore, although the computer science field is dominated by men and there is some evidence of a gender difference for spatial skills, it is unclear whether or not the actual process of learning coding differs by gender in young children.

2.3.2. Appeal. In discussing how children learn from educational media, Fisch (2004a) argues that if children do not enjoy a program, they simply won't turn it on, thus eliminating any potential educational benefit of the program. Many other researchers have also pointed to the importance of program appeal as a prerequisite for learning. Appeal refers to the degree to which children describe liking a program (Linebarger, McMenamin, & Wainwright, 2009). To that end, Linebarger and McMenamin (2010) assessed the role of program appeal in educational content comprehension. They found that when children liked a particular program, they were more likely to view the program repeatedly. This repetition was found to be a key component in learning content such as vocabulary words. Specifically, repetition mediated the relationship between appeal and comprehension such that greater appeal led to greater repetition of the program content, which in turn led to greater comprehension of the content. The study also found that emotional investments with onscreen characters lead to greater interest and stronger motivation to attend to the actions or events that involve these preferred characters (Linebarger & McMenamin, 2010).

Though much of the work on appeal has been conducted with educational television (e.g. Linebarger & McMenamin, 2010; Linebarger & Piotrowski, 2006), it is likely that these findings would hold true for other types of educational media as well. Therefore, it is possible that children who find coding apps more appealing would pay more attention to the apps and engage with them more often and more enthusiastically, leading to increased learning.

On the other hand, there is some reason to believe that appeal could lead to decreased learning in certain instances. The principle of narrative dominance (Fisch, 2000) posits that children pay attention to narrative and characters before attending to educational content. Therefore, very high levels of appeal for the characters may lead to decreased attention to and comprehension of the educational content. For example, Wakshlag, Reitz, and Zillman (1982) found that when an educational television program was accompanied by appealing music, thus increasing the overall appeal of the program, children paid less attention to the educational content and were less able to retrieve information from the program. In a more recent study by Aladé and Nathanson (2016), interest in the content (i.e. appeal) was not found to have any relationship with learning.

2.4. The Current Study

Much research on children's learning of coding have used qualitative research designs (e.g. Fessakis et al., 2012). While qualitative work offers rich, descriptive information, the current study seeks to further substantiate these rich descriptions with formal quantitative data reflecting children's learning. Specifically, the current study looks at children's ability to learn foundational coding skills through playing two commercially available apps, *Daisy the Dinosaur* and *Kodable,* in a classroom setting. We operationalize learning as (a) improvement in understanding basic coding commands, and (b) demonstrating successful app play, both measured using quantitative scales. Given previous research showing that young children can indeed learn coding skills (e.g., Kazakoff & Bers, 2014), we first predicted that:

H1: Young children (4- to 6-years old) will learn foundational coding skills through participation in a one-week program that teaches coding via guided app play on two coding apps

In order to transfer learning to new situations, it is necessary for children not only to learn the specific target skill, but also to develop a schema about the concept that they can apply to new situations (Barr, 2010; Fisch, Kirkorian, & Anderson, 2005). By playing with two different types of coding games, we thought that children in this particular coding camp might also develop a broader understanding of what coding is and how it can be used in other contexts. Therefore, this research explores whether children can gain a conceptual understanding of what coding is through their participation in the program. Our first research question therefore asked:

RQ1: Can children gain a conceptual understanding of what coding is through their participation in a one-week program that teaches coding through two coding apps?

Lastly, considering the disparity of women in STEM fields (Beede et al., 2011; Ceci, Williams, & Barnett, 2009; National Science Foundation, 2013) and the importance of appeal and interest in children's learning from media (e.g., Linebarger & McMenamin, 2010), our final research questions asked:

RQ2a: Does gender moderate children's learning to code via apps?

RQ2b: Does appeal moderate children's learning to code via apps?

3. Method

In order to better understand the nature and effectiveness of tablet app games that claim to teach coding to young children, we conducted a multi-method examination of a one-week summer program designed to teach coding to pre-kindergarten and kindergarten students via two coding apps. Two trained researchers acted as classroom assistants for the duration of the one-week camp, conducting the interviews and administering coding assessments with the children for the study. In addition to the classroom observation, parents gave us permission to access child standardized test scores and demographic information (child gender, ethnicity, and date of birth) from the summer camp office. All procedures were approved by the university's Institutional Review Board.

3.1. Participants

Participants (N = 28) were four, five, or six years old (M = 5.15) at the time of the study and enrolled in a half-day coding summer program for gifted and talented youth in Chicago, IL. Half of the participants were girls (n = 14). Nearly half identified as White/Caucasian (46%, n = 13), nine were Asian/Pacific Islanders (32%), one participant identified as Hispanic/Latino, and the rest were multiracial (n = 5). We obtained informed consent from the parents of all participants and verbal assent from all participating students. Two participants did not complete pre-test interviews and so only their post-test scores are including in analyses.

The gifted and talented program organizing these summer enrichment programs offers a host of classes in language arts, math, science, design and engineering, and computer science at several sites throughout the greater Chicagoland area. To enroll in the summer camps, students must score in the 90th percentile or above on a national standardized achievement test for their grade level in an area that is relevant to the course. The class observed for this study included a mix of students who scored in the 90th percentile or above on a national standardized achievement for math and/or language arts (e.g., Kaufman Test of Educational Achievement, Kaufman Brief Intelligence Test). We collected this information from camp administrators.

3.2. Coding Camp Curriculum

The course met for five days in the summer of 2016. There was a morning and an afternoon session with different participants. Our sample represents children from both morning and afternoon sessions. Each session was three hours long. Each day followed the same general schedule: students arrived, had a welcome circle to introduce the coding theme for the day (e.g., loops, repeat, if/then conditions), rotated through four different activity centers before having snack, a break outdoors, and then returned indoors to learn a new coding concept (e.g., troubleshooting). The day ended with one more rotation through activity centers and a closing circle. Every day, except for the first day, four tablets/iPads were set out at one center – the tablet center. The other three "centers" changed daily, but often included some combination of a writing/drawing center, block play, and an arts and crafts project.

The classroom teacher and classroom aide were trained and state-certified early childhood educators. The classroom teacher taught this same course the year prior and had taught the course at a different site the week prior to our observation. The teacher was very familiar with the apps used in the curriculum.

Over the course of the week-long curriculum, students were taught concepts of sequencing, conditions, and loops using two tablet-based apps: *Daisy the Dinosaur* and *Kodable*. According to the Apple App Store, *Daisy the Dinosaur* (hereafter referred to as *Daisy*) is an iOS app designed to teach children the basics of coding ("Daisy the Dinosaur," 2016). Using a drag and drop interface with written commands, the app aims to teach children sequencing, loops, and conditional events. In challenge mode, the app asks players to make Daisy, the animated dinosaur, accomplish a specific task using programming instructions. For example, the first instruction in challenge mode is to "make Daisy move to the star" where the star is several inches away from Daisy. Players must drag and drop the "move" command button listed on the left side of the screen into a large block box above Daisy to move the character (see Figure 1 for example of *Daisy* challenge mode). Students were told to either play in challenge mode (gameplay with specific instructions to complete) or free play mode while at the tablet table, depending on the day.

On day three of the program, *Kodable* was introduced to the classroom. According to the *Kodable* website, this app provides a full set of curricula designed to teach elementary students coding – first by working on foundational skills (ideal for kindergarten, first, and second grade) and then by actually reading and writing code for third, fourth, and fifth grades ("Kodable Curriculum," 2016). *Kodable* is another drag and drop interface using directional arrows as commands. At each level, a 'fuzz' character is presented at one end of a snake-like maze that extends from the left to the right side of the screen. Players must move the appropriate directional arrows (i.e. up, down, right, and left) into command boxes to help the 'fuzz' roll through this maze and onto the next level (see Figure 2 for example of *Kodable* sequencing level). By completing the mazes at each level, players learn about sequencing, loops, and conditions.

3.3. Measures

As part of the study, participants were video-recorded while they played at each station throughout the course of the camp. Before the start of camp on the first and last day, all participants also participated in brief interviews and gameplay with a research team member.

3.3.1. Pre-post interview. On the first and last day of the camp, participants completed a one-on-one interview with one of the researchers in a quiet space in the classroom. The interview was audio recorded, and participants were given pictorial response options (i.e. smiley face scale) whenever possible. The interview included questions about children's familiarity with various devices (e.g., tablets, computers, etc.), their familiarity with various coding apps (e.g., *Kodable*, *Daisy the Dinosaur*, *ScratchJr*, etc.), how much they liked *Daisy* and *Kodable*, and whether they could identify the *Daisy* sequencing commands.

Device familiarity. To get a sense of children's tech savvy-ness, participants were shown pictures of different technological devices (i.e., laptop, tablet, smartphone, digital camera) and asked to point to the devices they knew how to use. Participants received one point for each device they said they knew; points were summed to create a device familiarity score (0 to 4 points, pre-test M = 2.46).

App familiarity. To get a sense of their level of experience with coding-related apps, participants were shown images of 12 coding or coding-related apps, including the two apps used in the coding camp (i.e., *ScratchJr.*, *Superhero Comic Book Maker*, *Kodable*, *Scratch*, *Toontastic*, *Lightbot*, *Hopscotch*, *Doodlecast*, *Daisy the Dinosaur*, *Toca Builder*, *Cargo-Bot*, and *Minecraft*). Again, participants were asked to point to all the apps they knew how to use. Participants received one point for each app they said they knew; points were summed to create an app familiarity score (0 to 12 points, pre-test M = 1.65).

App appeal. We were interested in whether children's interest in the apps was related to their learning from them. At post-test, participants were asked how much they liked *Daisy the Dinosaur* and *Kodable*, which they had been playing in camp that week. Specifically, we asked children "How much do you like playing with [XX app]?" and "How much would you like to play with [XX app] again?" Children answered each of these questions using a four-point Likert-type scale (0-3), with response options being "not at all," (0), "a little," (1), "a lot," (2), or "a whole lot," (3). Scores for the two questions were summed to create a posttest app appeal score for each game ranging from 0-6 (M_{Daisy} = 4.48, SD = 1.76; $M_{Kodable}$ = 5.46, SD = 0.83).

Daisy command knowledge. Participants were shown pictures of four of the command buttons in *Daisy the Dinosaur* (i.e., "when", "move", "grow", and "jump") and asked to verbally tell the researcher what each button means or does. At pre-test this represented a baseline measure of participants' familiarity with and knowledge of how to play *Daisy*. At post-test this was a direct measure of how much they learned about *Daisy the Dinosaur* as a result of participation in the coding camp. Responses were coded as incorrect (0) or correct (1), resulting in a composite score ranging from 0-4. On average, children knew 0.5 buttons (*SD* = 1.14) at pre-test and 2.3 buttons (*SD* = 2.68) at post-test. We subtracted scores at pre-test from scores at post-test to create a "Daisy Learning Score."

"What is Coding?" Participants were asked to explain in their own words, what coding is. The researcher asked each child, "Can you tell me what coding is?" and, if needed, used the prompt, "What do you do when you are coding something?" Participants' responses were recorded verbatim for later coding.

After reviewing all verbatim responses, researchers created a four-point coding scheme to discriminate quality of verbal responses. Responses received a score of "3" if they explained a cause and effect sequence that could be applied to technology broadly. For example, one child noted, "coding is controlling something in a way, you put in a code for something to do something." Responses received a "2" if they explained a cause and effect sequence that can only be applied to the target apps. An example from our participants was, "you need to put arrows in the boxes and then the little ball moves." Responses received a "1" if they mentioned a feature of either of the target apps, but did not mention a causal or action sequence. For instance, one child said, "you code with Daisy the Dinosaur." Other incorrect or "don't know" responses received a "0". After the coding scheme was established, two coders who had not seen the responses used the coding manual to give a numerical score for each response at pre-test and post-test. Interrater reliability was high (κ = .90).

3.3.2. Pre-post gameplay assessment. Participants were video-recorded playing up to four different levels of the *Kodable* game on the first (pre-test) and last (post-test) day of the camp. The four levels we tested were selected a priori: two in sequencing, one in conditions, and one in loops. These specific tasks were chosen because they were (a) feasible to complete in the three-minute time window and (b) the second challenge at each foundation skill level that was available with our subscription of the product. The researcher gave one basic instruction to each child (i.e., “For this game, you have to drag the arrows over to the boxes to help the little guy get through the maze”) and then asked each child, “Can you show me how you play?” If the child succeeded at a level, the researcher would advance the game to the next level, until three total minutes had elapsed.

We designed a coding scheme to track the levels children completed during the gameplay assessment as well as their performance quality at each level. Adapted from Bers, Flannery, Kazakoff, and Sullivan (2014) coding scheme, two trained coders rated gameplay on a six-point ordinal-like scale for each level played (four levels possible in total) where “0” was assigned to those who did not attempt gameplay and “5” was assigned for successful completion of the level. For example, a student who, at pre-test, successfully passed level one and only tried to attempt level two, but was completely off target would receive a score of 2.1. See Appendix A for full coding scheme.

To quantify improvement from pre- to post-test, a difference score was created by subtracting pre-test highest score from post-test highest score. Of the 56 total videos, two coders double-coded 14 of them (7 pre-test and 7 post-test, 25%), and the rest were split between the two coders. Interrater reliability was good ($\kappa = .75$).

4. Results

First, we checked for potential covariates by running correlation analyses. Age, standardized test percentiles, device familiarity, and app familiarity were not significantly correlated with any of our dependent variables (i.e., *Daisy* command knowledge, *Kodable* gameplay, “What is Coding” scores) and thus were not included in the following analyses.

4.1. Learning and Articulating Foundational Coding Skills

To address our first hypothesis, we ran paired-samples t-tests to compare children's command knowledge scores and gameplay scores from pre-test to post-test. Participants significantly improved in their knowledge of *Daisy* commands from pre-test (M = 0.5, SD = 1.14) to post-test (M = 1.88, SD = 1.42); $t(26) = -4.80$, $p < .001$. Specifically, significantly more students were able to correctly explain the three sequencing commands at post-test (i.e. "move," "jump," "grow"), all $p < .001$, but there was no difference in successfully explaining a condition command ("when") at post-test (see Table 1 for t-test results).

In terms of *Kodable* gameplay, participants were able to progress significantly farther and score significantly higher in *Kodable* gameplay from pre-test (M = 1.62; SD = 0.81) to posttest (M = 2.99; SD = 0.91). The average gameplay difference score was positive (M = 1.36; SD = 0.77; Range = 1.1-4.5); $t(27) = -9.42$, $p < .001$. Descriptively speaking, only 32% of participants were able to successfully complete the first level of the game (sequencing) at pre-test compared to almost all participants (92%) at post-test. Taken together, these findings support our first hypothesis; students were able to learn foundational coding skills using these two apps.

We were also interested in children's explanation of coding, so we used a paired samples t-test to compare "What is Coding?" scores from pre-test to post-test. Scores on the "What is Coding?" assessment did not significantly improve from pre-test (M = 0.48, SD = .85) to post-test (M = 0.74, SD = .90); $t(27) = -1.23$, $p = 0.23$). Students did not improve in their ability to verbally articulate general understanding of the concept of coding as measured by our open-ended question and coding scheme.

4.2. Gender, App Appeal, and Coding Knowledge

To address RQ2a (whether gender predicts children's learning to code via apps), we first ran an ANOVA to compare *Daisy* command knowledge difference scores by gender. There was no difference in these scores by gender, $F(1, 24) = 1.84$, $p = .188$. We ran another ANOVA to compare the *Kodable* gameplay difference scores by gender; again, there was no significant difference ($F(1,26) = 3.03$, $p = .094$), however the mean score for girls trends higher. There was also no significant difference in "What is Coding?" scores by gender ($F(1,25) = .05$, $p = .83$).

In terms of appeal, at post-test, all of the participants liked *Kodable* "a lot" or "a whole lot." Students liked *Kodable* ($M = 5.46$, $SD = .84$, Range = 4-6) more than *Daisy* ($M = 4.48$, $SD = 1.76$, Range = 1-6), $t(27) = -2.56$, $p = .017$, but reported they would want to play both again. Compared to the coding app that they previously knew best ($M = 4.95$, SD = 1.07), students liked *Daisy* about the same, $t(27) = 1.46$, p = .155, but, on average, liked *Kodable* slightly, but not quite significantly more, $t(27) = -1.94$, $p = .063$. Overall, appeal for both apps was quite high ($M = 10$, $SD = 1.75$, Range = 7-12).

To answer RQ2b, we ran a regression analysis using the *Daisy* command knowledge difference score as the dependent variable and *Daisy* appeal score as the independent variable. *Daisy* appeal was a significant predictor of improvement in *Daisy* command knowledge; $F(1,23) = 6.57$, p = .017, such that as *Daisy* appeal scores increase, on average, *Daisy* command knowledge increases. Then, we ran a linear regression analysis with the *Kodable* gameplay difference score as the dependent variable and *Kodable* appeal score as the independent variable. *Kodable* appeal was not a significant predictor of improvement in Kodable gameplay, $F(1, 26) = 1,65$, $p = .210$. Appeal for either app was not related to "What is Coding" scores (*Daisy*: $F(1,24) = 2.36$, $p = .14$; *Kodable*: $F(1,25) = .15$, $p = .70$).

5. Discussion

This study utilized a multi-method design, including both child interviews and gameplay assessments, to test the effectiveness of a camp-based program that uses two commercially available apps to teach foundational coding skills to young children. In sum, participants improved in their knowledge of *Daisy* commands and in their *Kodable* gameplay over the course of the week-long camp, suggesting that they were in fact able to learn foundational coding skills via structured app play. This finding was not surprising as it echoes previous work that found pre-school and early elementary school-age children are able to learn sequencing over several training sessions (e.g. Flannery & Bers, 2013). However, it adds to the existing body of literature on this topic by supplementing previous qualitative descriptions with quantitative data on children's learning of coding skills. Moreover, this research adds coding to the growing list of foundational STEM concepts that children can learn in the preschool years, and situates tablet apps as being a potentially powerful way of exposing children to basic technology concepts. It also adds to the literature on two specific games, *Kodable* and *Daisy*, which have not been evaluated systematically.

Although this study was designed to measure young children's learning of programming skills specifically, it is also relevant to the broader literature on preschoolers' development of computational thinking skills. Computational thinking has been defined as a set of problem-solving skills and other concepts commonly used in computer science (Bers et al., 2014; Computer Science Teachers Association, 2011; Mannila et al., 2014). Our findings are in line with other studies on preschoolers' learning of computational thinking skills in that the participants improved in their programming performance after advancing through a dedicated coding curriculum (Bers et al., 2014; Sullivan & Bers, 2015). Furthermore, this improvement was best demonstrated with the activities they spent the majority of their time practicing (i.e.

gameplay). Bers et al. (2014) found that kindergarteners initially improved in their debugging, correspondence, sequencing, and control flow skills (identified as elements of computational thinking), but then decreased in their achievement of these activities subsequently. The researchers posit that this decline in performance was likely related to the decreased amount of time teachers spent on more advanced lessons and the fact that the tasks became incrementally more challenging (Bers et al., 2014). Our findings tell a similar story; participants improved in *Kodable* gameplay and most of *Daisy* command knowledge, but not in successfully explaining the "when" command in *Daisy* or describing 'coding' more generally. Together, these two studies suggest that when guiding children through coding or computational thinking curriculum, as much time needs to be spent on the more advanced skills as on the basic foundational skills.

This study further extends the literature by exploring child characteristics that could potentially influence learning, namely gender and appeal. Contrary to some research that has found gender differences in coding-related skill acquisition (Nourbakhsh, Hamner, Crowley, & Wilkinson, 2004), but in line with more recent research (Papadakis, Kalogiannakis, & Zaranis, 2016; Sullivan & Bers, 2016), we did not find any gender differences in our study. We see this finding as encouraging evidence for girls' participation in computer science and programming experiences. The classroom observed in this study was almost exactly equal in gender representation. This, in and of itself, is noteworthy, as the camp was completely opt-in based, and children (and their parents) had many choices of course topics to enroll in for that week. The lack of observed gender difference supports the idea that girls should be exposed to STEM as early as possible since there seems to be no underlying cognitive difference to blame for their lack of participation in the STEM pipeline. It is important to note that our participants were highly gifted, and therefore more research is needed to confirm whether this lack of gender difference would be found in a more generalizable sample.

While gender was not found to be an important moderator, we found that appeal *was* significantly related to learning for one of the two apps used in this study. This relationship makes sense given previous work on appeal in educational media (i.e. Linebarger & McMenamin, 2010; Linebarger & Piotrowski, 2006). However, our results are noteworthy because they suggest the appeal of apps may have a more nuanced effect on children's learning from apps (Fisch, 2004). Children in our study scored high in appeal toward the apps, and we even observed a ceiling effect on appeal for *Kodable*. Nonetheless, generally appealing apps did not result in all children learning equally from the them. In other words, our results suggest that embedding coding in an app does not inherently make it appealing; variability in app appeal – even at high levels of appeal – can lead to variations in learning for young children.

Of course, it should be noted that appeal may also be affected by various factors like the particular presentation of the content as well as peer and adult feedback around the content. Although we followed well-established procedures from earlier research (e.g. Aladé & Nathanson, 2016; Meyer & Skora, 2010), we recognize these factors may have influenced children's interest in the games, and recommend future research use both quantitative and qualitative approaches to piece apart whether specifics about the game or about the context in which it is played impacts children's learning. Regardless, our finding about appeal is noteworthy for media producers, parents, and educators to keep in mind: while quality of the curriculum is certainly important, when it comes to apps, maximizing the appeal of the apps may be imperative to children's optimal learning from them.

In addition to learning the skills taught in the apps, one important question we had was whether children would be able to gather a broader understanding of the concept of coding after participating in this particular summer program. We did not see evidence of an increase in broader understanding. This was a very exploratory question, and eliciting open-response may have limited our ability to gauge children's actual knowledge of coding. It is possible that while children did gain a broader understanding of coding, they still lack the vocabulary and verbal skills needed to describe the concept. Indeed, other empirical work with a similar age group also found that children failed to demonstrate learning from media when given open-ended prompts like the ones used in this study (e.g. Mares & Acosta, 2008). Future research should continue to look not only at basic skill acquisition, but also seek to develop standardized measures of conceptual coding knowledge for this age group.

6. Conclusion

Overall, these findings demonstrate that digital apps can indeed be used to teach young children foundational coding skills and provide a fun and enjoyable platform on which to practice these skills. Despite these important findings, this research was not without limitations. For one thing, this group of participants had self-selected (by way of parents or guardians) to participate in the particular program that we studied. Thus, they likely started out at higher levels of interest in coding than non-participants. Additionally, students in this class were not representative of other children their age as they were required to be of above average intelligence for entry into the program. While this research is an important addition to the literature on young children and programming, we should be wary of extrapolating findings from this research to general populations of pre-kindergarten and kindergarten students. Lastly, we must reiterate that children in this program were taught through a relatively structured curriculum, where target skills were demonstrated by the teacher before individual app play, and individual app play was closely guided and monitored by the teacher and teaching assistants. Therefore, we cannot generalize our findings to say that merely playing with a coding app independently will produce the observed effects.

Our research provides new quantitative data to supplement the qualitative data on this phenomenon, but further research is needed to determine if children can transfer these skills to a novel programming language or other related context. Future research should be designed to gain a clearer understanding of whether independent gameplay can support learning of foundational coding skills, and how parents and teachers can effectively engage with children around coding to promote children's learning from guided instruction and modeling. Moreover, there is reason to believe that the appeal of the apps is one of their greatest features, as this familiarity and enjoyment could lead children to play more games like these that eventually might foster a love of STEM learning later in life. If these positive early experiences with coding and programming are enjoyable and effective, perhaps they might facilitate the training of more computer scientists and eventually enable us to meet the demand for skilled STEM workers.

References

Aladé, F., Lauricella, A. R., Beaudoin-Ryan, L., & Wartella, E. (2016). Measuring with Murray: Touchscreen technology and preschoolers' STEM learning. *Computers in Human Behavior, 62*, 433-441.

Aladé, F., & Nathanson, A. I. (2016). What preschoolers bring to the show: The relation between viewer characteristics and children's learning from educational television. *Media Psychology*, 1-25.

Anderson, D. R., Lavigne, H. J., & Hanson, K. G. (2013). The Educational Impact of Television. In E. Scharrer (Ed.), *The International Encyclopedia of Media Studies* (1 ed., Vol. 5). Hoboken, NJ: Blackwell Publishing Ltd.

Ball, S., & Bogatz, G. A. (1970). *The First Year of Sesame Street: An Evaluation: a Report to the Children's Television Workshop*. Retrieved from Princeton, NJ:

Barr, R. (2010). Transfer of learning between 2D and 3D sources during infancy: Informing theory and practice. *Developmental Review, 30*, 128-154.

Beede, D., Julian, T., Langdon, D., McKittrick, G., Khan, B., & Doms, M. (2011). *Women in STEM: A Gender Gap to Innovation*. Washington, DC.

Bers, M. U. (2012). *Designing digital experiences for positive youth development: From playpen to playground*. New York, NY: Oxford University Press.

Bers, M. U. (2017). Why Kids Should Code. Retrieved from http://now.tufts.edu/articles/manifesto-kids-code?utm_source=Tufts+Now+-+Faculty+and+Staff&utm_campaign=34cb1424ff-Tufts_Now_external_170927&utm_medium=email&utm_term=0_e2c82ed1e3-34cb1424ff-207435605

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education, 72*, 145-157. doi:10.1016/j.compedu.2013.10.020

Ceci, S. J., & Williams, W. M. (2011). Understanding current causes of women's underrepresentation in science. *Proc Natl Acad Sci U S A, 108*(8), 3157-3162.

Ceci, S. J., Williams, W. M., & Barnett, S. M. (2009). Women's underrepresentation in science: Sociocultural and biological considerations. *Psychological Bulletin, 135*(2), 218-261.

Clements, D. H., & Gullo, D. F. (1984). Effects of Computer Programming on Young Children's Cognition. *Journal of Educational Psychology, 76*(6), 1051-1058.

Common Sense Media. (n.d.). Cool Coding Apps and Websites for Kids. Retrieved from https://www.commonsensemedia.org/lists/coding-apps-and-websites

Computer Science Teachers Association. (2011). *Operational Definition of Computational Thinking*. Retrieved from http://www.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf.

Daisy the Dinosaur. (2016). Retrieved from https://itunes.apple.com/us/app/daisy-the-dinosaur/id490514278?mt=8

Etta, R., Kirkorian, H. L., & Choi, K. (2017). *Preschoolers' Learning From Children's Books: Effects of Platform and Interactivity*. Paper presented at the International Communication Association, San Diego, CA.

Fayer, S., Lacey, A., & Watson, A. (2017). *STEM Occupations: Past, Present, And Future*. Washington, D.C.: U.S. Bureau of Labor Statistics.

Fessakis, G., Gouli, E., & Mavroudi, E. (2012). Problem solving by 5-6 years old kindergarten children in a computer programming environment. *Computers & Education, 63*, 87-97.

Fisch, S. M. (2000). A capacity model of children's comprehension of educational content on television. *Media Psychology, 2*(1), 63-91.

Fisch, S. M. (2004a). *Children's learning from educational television*. Mahwah, NJ: Lawrence Erlbaum.

Fisch, S. M. (2004b). *Children's learning from educational television: Sesame Street and beyond*. Malweh, NJ: Routledge.

Fisch, S. M., Kirkorian, H. L., & Anderson, D. R. (2005). Transfer of learning in informal education: The case of television. In J. P. Mestre (Ed.), *Transfer of Learning from a Modern Multidisciplinary Perspective*: Information Age Publishing.

Flannery, L., & Bers, M. U. (2013). Let's dance the "robot hokey pokey!": Children’s Programming Approaches and Achievement

throughout Early Cognitive Development. *Journal of Research on Technology in Education, 46*(1), 81-101.

Flannery, L., Kazakoff, E. R., Bonta, P., Silverman, B., Bers, M. U., & Resnick, M. (2013). *Designing ScratchJr: Support for Early Childhood Learning Through Computer Programming.* Paper presented at the Interational Conference on Instruction Design and Children, New York, NY.

García-Peñalvo, F. J., Rees, A. M., Hughes, J., Jormanainen, I., Toivonen, T., & Vermeersch, J. (2016). *A survey of resources for introducing coding into schools.* Paper presented at the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality Salamanca, Spain.

Golden, T., Blackburn, J. M., Scarlato, M., Free, J. U., & Degelman, D. (1986). Concept Learning in Preschool Children: Effects of a Short-Term Logo Experience. *Journal of Educational Computing Research, 2*(2), 199-205. doi:10.2190/rh2k-4aq7-2598-tvea

Hirsh-Pasek, K., Zosh, J. M., Golinkoff, R. M., Gray, J. H., Robb, M. B., & Kaufman, J. (2015). Putting education in "educational" apps: lessons from the science of learning. *Psychological Science in the Public Interest, 16*(1), 3-34.

Huber, B., Tarasuik, J., Antoniou, M. N., Garrett, C., Bowe, S. J., & Kaufman, J. (2016). Young children's transfer of learning from a touchscreen device. *Computers in Human Behavior, 56*, 56-64.

Karadeniz, S., Samur, Y., & Ozden, M. Y. (2014). *Playing with algorithms to learn programming: A case study on 5 years old children.* Paper presented at the International Conference on Information Technology and Applications, Sydney, Australia.

Kazakoff, E. R., & Bers, M. U. (2014). Put Your Robot In, Put Your Robot Out: Sequencing through Programming Robots in Early Childhood. *Journal of Educational Computing Research, 50*(4), 553-573. doi:10.2190/EC.50.4.f

Kazakoff, E. R., Sullivan, A., & Bers, M. U. (2012). The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. *Early Childhood Education Journal, 41*(4), 245-255.

Kelly, B. (2012). The state of STEM and jobs. Retrieved from https://www.usnews.com/news/articles/2012/09/21/the-state-of-stem-and-jobs

Kodable Curriculum. (2016). Retrieved from https://www.kodable.com/curriculum

Krcmar, M., & Cingel, D. P. (2014). Parent–child joint reading in traditional and electronic formats. *Media Psychology, 17*(3), 262-281.

Linebarger, D. L., & McMenamin, K. (2010). *Between the Lions Mississippi literacy initiative: 2008-2009 review*. A final report prepared for Corporation for Public Broadcasting. Philadelphia, PA: University of Pennsylvania, Annenberg School for Communication.

Linebarger, D. L., McMenamin, K., & Wainwright, D. K. (2009). *Summative evaluation of Super Why!* Philadelphia, PA: Annenberg School for Communication.

Linebarger, D. L., & Piotrowski, J. T. (2006). *Pinky Dinky Doo: Evaluating the educational impact and appeal of Pinky Dinky Doo on preschool children*. A final report to Sesame Workshop. Philadelphia, PA: Annenberg School for Communication.

Linn, M. C., & Petersen, A. C. (1985). Emergence and characterization of sex differences in spatial ability: A meta-analysis. *Child Development, 56*, 1479-1498.

Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L., & Settle, A. (2014). *Computational Thinking in K-9 Education.* Paper presented at the Proceedings of the working group reports of the 2014 on innovation & technology in computer science education conference, Uppsala, Sweden.

Mares, M.-L., & Acosta, E. E. (2008). Be Kind to Three-Legged Dogs: Children's Literal Interpretations of TV's Moral Lessons. *Media Psychology, 11*(3), 377-399.

Meyer, S. E., & Skora, I. (2010). *A Summative Evaluation of the Impact and Appeal of Super WHY! Reading Camp.* Jackson Heights, NY: Paradigm Research & Consulting

Moore, D. S., & Johnson, S. P. (2008). Mental rotation in human infants: a sex difference. *Psychol Sci, 19*(11), 1063-1066. doi:10.1111/j.1467-9280.2008.02200.x

Morgado, L., Cruz, M., & Kahn, K. (2010). Preschool cookbook of computer programming topics. *Australasian Journal of Educational Technology, 26*(3), 309-326.

National Center for Women & Information Technology. (2017). By the numbers. Retrieved from https://www.ncwit.org/sites/default/files/resources/btn_03232017_web.pdf

National Science Foundation. (2013). *Women, minorities, and persons with disabilities in science and engineering: 2013*. Arlington, VA.

Nourbakhsh, I. R., Hamner, E., Crowley, K., & Wilkinson, K. (2004). *Formal Measures of Learning in a Secondary School Mobile Robotics Course.* Paper presented at the IEEE International Conference on Robotics and Automation (ICRA), New Orleans, LA.

O'Dell, J. (2013). Why your 8-year-old should be coding. Retrieved from https://venturebeat.com/2013/04/12/why-your-8-year-old-should-be-coding/

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), 187-202. doi:10.1504/ijmlo.2016.077867

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

Petersen, J., & Hyde, J. S. (2014). Gender-related academic and cccupational interests and goals. In S. L. Lynn & S. B. Rebecca (Eds.), *Advances in Child Development and Behavior* (Vol. 47, pp. 43-76). San Diego, CA: Elsevier.

Piotrowski, J. T., & Valkenburg, P. M. (2015). Finding orchids in a field of dandelions: Understanding children's differential susceptibility to media effects. *American Behavioral Scientist, 59*(14), 1776-1789.

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

Prensky, M. (2008). Programming is the new literacy. Retrieved from https://www.edutopia.org/literacy-computer-programming

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

Rothwell, J. (2014). Short on STEM talent. Retrieved from https://www.usnews.com/opinion/articles/2014/09/15/the-stem-worker-shortage-is-real

Schroeder, E. L., & Kirkorian, H. L. (2016). When seeing is better than doing: Preschoolers' transfer of STEM skills using touchscreen games. *Frontiers in Psychology, 7*, 1-12.

Siegle, D. (2013). iPads: Intuitive Technology for 21st-Century Students. *Gifted Child Today, 36*(2), 146-150.

Strawhacker, A., Lee, M., Caine, C., & Bers, M. U. (2015). *ScratchJr demo: A coding language for Kindergarden.* Paper presented at the International Conference on Interaction Design and Children, Boston, MA.

Sullivan, A., & Bers, M. U. (2015). Robotics in the early childhood classroom: Learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade. *International Journal of Technology and Design Education, 26*(1), 3-20.

Sullivan, A., & Bers, M. U. (2016). Girls, Boys, and Bots: Gender Differences in Young Children’s Performance on Robotics and Programming Tasks. *Journal of Information Technology Education: Innovations In Practice, 15*, 145-165.

Sullivan, A., Kazakoff, E. R., & Bers, M. U. (2013). The wheels on the bot go round and round: Robotics curriculum in pre-kindergarten. *Journal of Information Technology Education: Innovations In Practice, 12*, 203-219.

U.S. Department of Education. (2015). A New Chapter for Computer Science Education. Retrieved from https://innovation.ed.gov/2015/12/07/a-new-chapter-for-computer-science-education/

U.S. Department of Education. (2016). New website highlights progress in early STEM education. Retrieved from https://innovation.ed.gov/2016/09/15/new-website-highlights-progress-early-stem-education/

U.S. Department of Education. (n.d.). Science, Technology, Engineering and Math: Education for Global Leadership. Retrieved from https://www.ed.gov/stem

Valkenburg, P. M., & Peter, J. (2013). The Differential Susceptibility to Media Effects Model. *Journal of Communication*, *63*(2), 221-243.

Voyer, D., & Voyer, S. D. (2014). Gender differences in scholastic achievement: a meta-analysis. *Psychol Bull*, *140*(4), 1174-1204. doi:10.1037/a0036620

Voyer, D., Voyer, S. D., & Bryden, M. P. (1995). Magnitude of sex differences in spatial abilities: A meta-analysis and consideration of critical variables. *Psychological Bulletin, 117*(2), 250-270.

Wai, J., Lubinski, D., & Benbow, C. P. (2009). Spatial ability for STEM domains: Aligning over 50 years of cumulative psychological knowledge solidifies its importance. *Journal of Educational Psychology, 101*(4), 817-835.

Wakshlag, J. J., Reitz, R. J., & Zillman, D. (1982). Selective Exposure to and Acquisition of Information From Educational Television Programs as a Function of Appeal and Tempo of Background Music. *Journal of Educational Psychology, 74*(5), 666-677.

Wang, M. T., Eccles, J. S., & Kenny, S. (2013). Not lack of ability but more choice: individual and gender differences in choice of careers in science, technology, engineering, and mathematics. *Psychological Science, 24*(5), 770-775. doi:10.1177/0956797612458937

Williams, W. M., & Ceci, S. J. (2012). When scientists choose motherhood. *American Scientist, 100*(2), 138-145.

Table 1

Means and standard deviations for *Daisy* command knowledge scores

Variables		*n*	Mean	SD	*t*	*p*
Move						
	Pre	26	.12	.33	-4.28	<.001
	Post	28	.54	.51		
Grow						
	Pre	26	.15	.37	-4.63	<.001
	Post	28	.62	.50		
Jump						
	Pre	26	.15	.37	-4.37	<.001
	Post	28	.65	.49		
When						
	Pre	26	.08	.27	0	1.00 ns
	Post	28	.08	.27		

Notes. SD = Standard deviation, ns = not significant

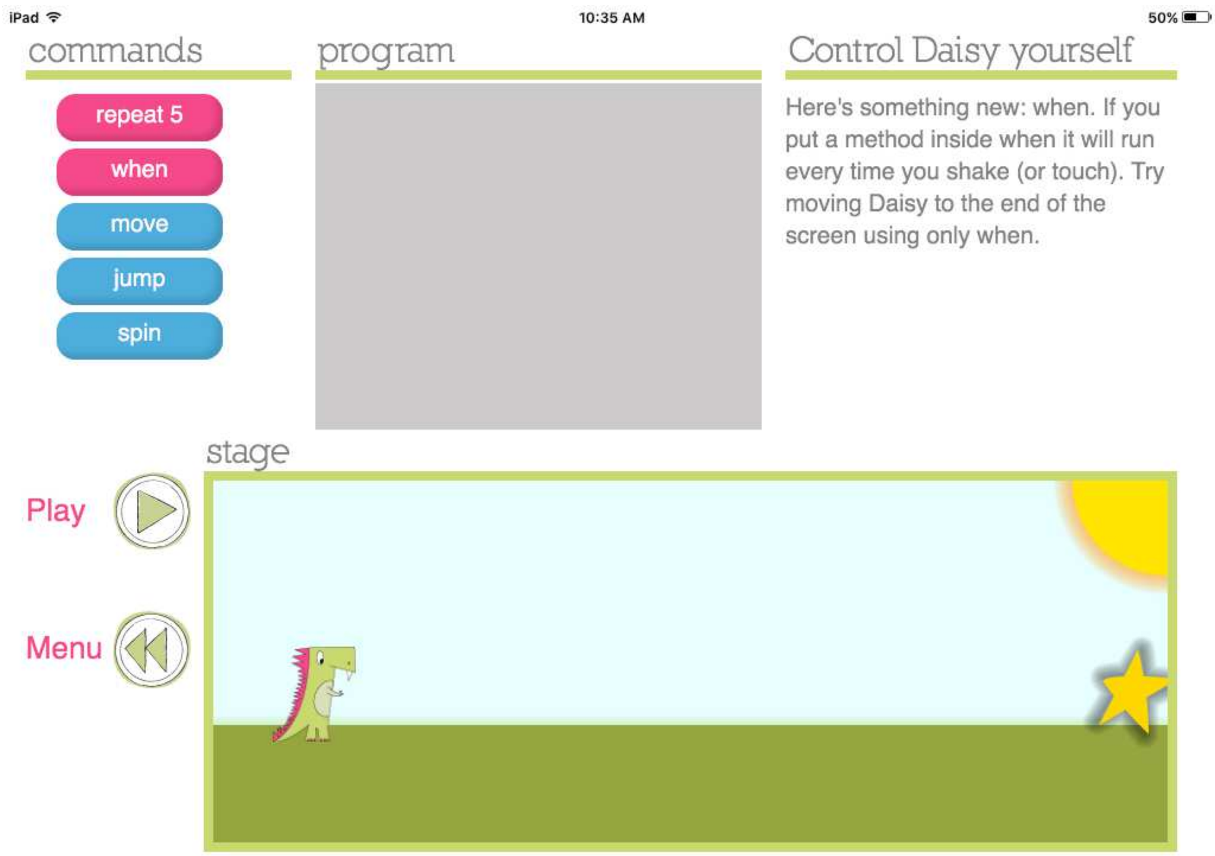

Figure 1. Screenshot of *Daisy the Dinosaur* in challenge mode.

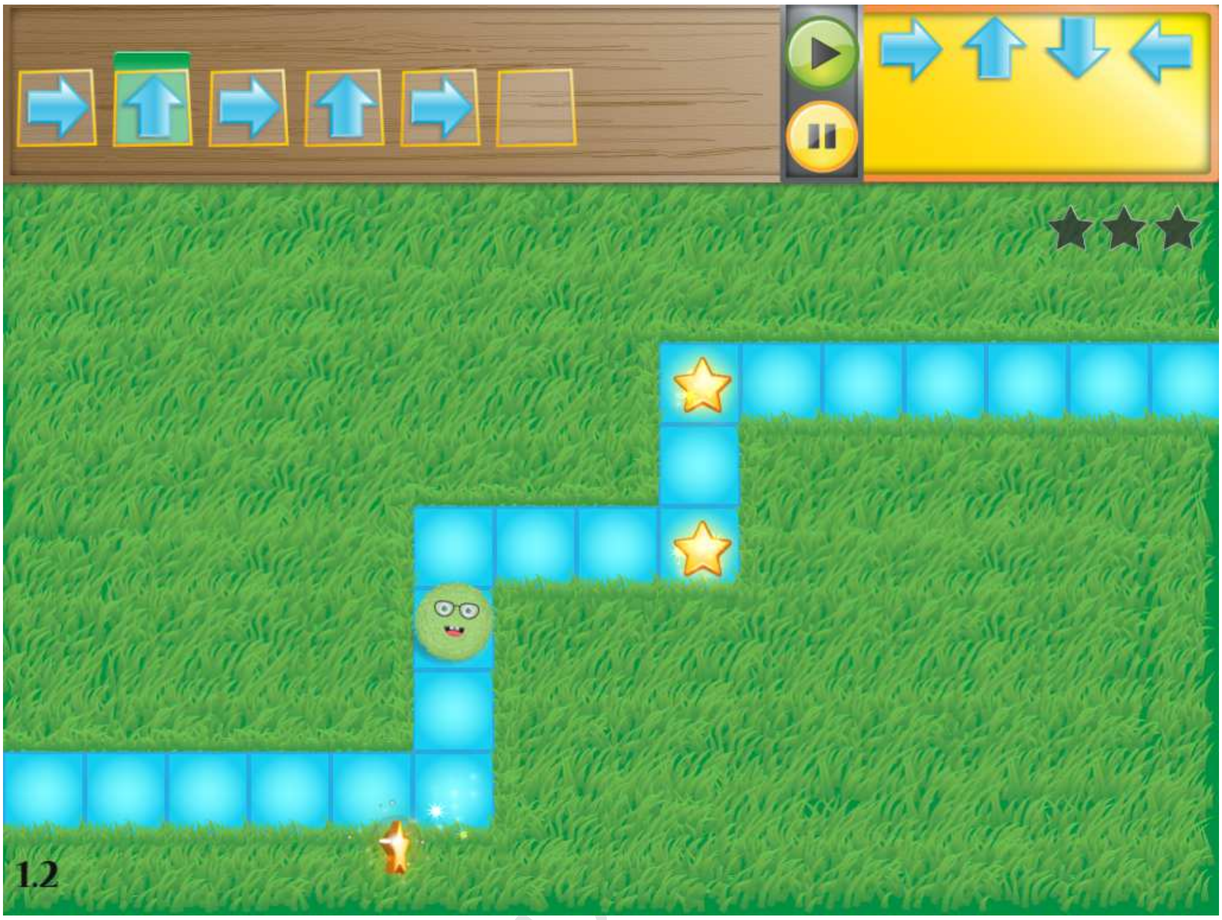

Figure 2. Screenshot of *Kodable* in sequencing game.

Highlights

- Children's learning of foundational coding skills via tablet apps was tested
- After one week, children demonstrated learning of game-specific coding skills
- Child gender was not related to learning outcomes
- App appeal was positively related to learning outcomes