Int J Technol Des Educ
DOI 10.1007/s10798-015-9304-5

Robotics in the early childhood classroom: learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade

Amanda Sullivan[1] · **Marina Umaschi Bers**[1]

Accepted: 9 March 2015

Abstract In recent years there has been an increasing focus on the missing "T" of technology and "E" of engineering in early childhood STEM (science, technology, engineering, mathematics) curricula. Robotics offers a playful and tangible way for children to engage with both T and E concepts during their foundational early childhood years. This study looks at N = 60 children in pre-kindergarten through second grade who completed an 8-week robotics curriculum in their classrooms using the KIWI robotics kit combined with a tangible programming language. Children were assessed on their knowledge of foundational robotics and programming concepts upon completion of the curriculum. Results show that beginning in pre-kindergarten, children were able to master basic robotics and programming skills, while the older children were able to master increasingly complex concepts using the same robotics kit in the same amount of time. Implications for developmentally appropriate design of technology, as well as structure and pace of robotics curricula for young children are addressed.

Keywords Robotics · Programming · Engineering · Education · Early childhood

Introduction

Research and policy changes over the past 5 years have brought about a newfound focus on STEM (science, technology, engineering, and math) education for young children (Sesame Workshop 2009; White House 2011). In particular, the "T" of technology and the "E" of engineering, which have been most neglected in early education, has gotten significantly more attention with the release of new learning standards and best practices for integrating

✉ Amanda Sullivan
amanda.sullivan@tufts.edu

[1] DevTech Research Group, Eliot Pearson Department of Child Study and Human Development, Tufts University, 105 College Ave, Medford, MA 02155, USA

Published online: 22 March 2015

technology into early childhood education (Barron et al. 2011; International Society for Technology in Education (ISTE) 2007; NAEYC and Fred Rogers Center for Early Learning and Children's Media 2012; U.S. Department of Education 2010).

As part of this push to teach technology and engineering to young children in a developmentally appropriate way, robotics and computer programming initiatives are growing in popularity amongst early childhood researchers and educators (Bers 2008). Recent work has shown how the field of robotics holds special potential for early childhood classrooms by facilitating cognitive as well as fine motor and social development (Bers et al. 2013). Young children can become engineers by playing with motors and sensors as well as storytellers by creating and sharing personally meaningful projects that react in response to their environment (Bers 2008).

The purpose of this study is to describe how robotics and computer programming can be used in early childhood classrooms and what young children (pre-kindergarten through second grade) can learn with these tools. This study describes an 8-week robotics curriculum at an urban public school and presents data collected on children's knowledge of foundational robotics and programming concepts after completing the curriculum.

Literature review

The missing "T" and "E" in STEM in early education

Although young children are growing up in an increasingly digital environment, school curriculum does not always focus on exploring the digital world until later elementary years. Only a small number of countries and regions (such as the United Kingdom) have established clear policies and frameworks for introducing technology to young children (Siu and Lam 2003). Instead, science curricula in early childhood classrooms are more likely to focus on the natural world including plants, animals, and the weather. While learning about the natural world is important, developing children's knowledge of the human-made world, the world of technology and engineering, is also needed for children to understand the environment they live in (Bers 2008).

In a typical early childhood classroom, it is not unusual to see young children exploring foundational engineering concepts by building and designing with crafts, recycled materials, and LEGO® pieces. However, what is unique to our world today is the fusion of electronics with mechanical structures. Children encounter sensors whenever they use a sink with automatic water dispensers or walk into a room where the lights come on without a switch (Bers and Horn 2010). Humans live in a world in which bits and atoms are increasingly integrated; however, we do not always teach our young children about this (Bers 2008). Robotics offers a way to teach young children about the types of sensors and electronics they encounter in daily life in a hands-on and engaging way. Teaching foundational programming concepts, along with robotics, makes it possible to introduce children to important ideas that inform the design of many of the everyday objects they interact with.

Kindergarten coders: robotics and programming for young children

Prior research suggests that children as young as 4 years old can successfully build and program simple robotics projects while learning a range of engineering and robotics concepts in the process (Bers et al. 2002; Cejka et al. 2006; Perlman 1976; Wyeth 2008; Sullivan et al. 2013). Research also suggests that children who are exposed to STEM curriculum and computer programming at an early age demonstrate fewer gender-based stereotypes regarding STEM careers (Metz 2007; Steele 1997) and fewer obstacles entering technical fields (Madill et al. 2007; Markert 1996). Based on this type of research, some countries, like the United Kingdom, have made learning to program and solve computational problems a required curricular framework (Department for Education 2013). However, in the United States, where the study described in this paper takes place, there are not yet any curricular frameworks requiring robotics or computer programming instruction.

In addition to teaching concepts related to technology and engineering, using robotics and computer programming in early childhood education can support the development of a range of cognitive and social milestones. For example, early studies with the text-based language Logo, have shown that computer programming can help young children with variety of cognitive skills, including number sense, language skills, and visual memory (Clements 1999). Educational robotics kits have become a new generation of learning manipulatives that help children develop a stronger understanding of mathematical concepts such as number, size, and shape in much the same way that traditional materials like pattern blocks, beads, and balls do (Resnick et al. 1998; Brosterman 1997). Unlike many apps and educational software developed for children, robotics activities do not involve sitting alone, in front of a computer. Rather, robotic manipulatives allow children to develop fine motor skills and hand-eye coordination while also engaging in collaboration and teamwork (Lee et al. 2013). Finally, robotics and computer programming curriculum also have the potential to foster "computational thinking" in young children. This term has been defined in many ways and encompasses a broad of analytic and problem-solving skills, dispositions, habits, and approaches used in computer science (Barr and Stephenson 2011; International Society for Technology Education and The Computer Science Teachers Association 2011; Lee et al. 2011). Computational thinking involves the ability to abstract from computational instructions (programming languages) to computational behaviors and to identify potential "bugs" and places for errors. (Wing 2006).

Some researchers argue that learning how to think like a computer scientist should be a mandatory skill that complements the "three Rs" (reading, writing, and arithmetic).

In the same way that the printing press facilitated the spread of the three Rs, computers can facilitate the spread of computational thinking (Wing 2006). Furthermore, when children learn a programming language, they are not "just learning to code, they are coding to learn" (Resnick 2013, pp. 5). They are solving problems in systematic ways, struggling to learn new powerful ideas, and expressing themselves with a variety of computational media.

Robotics kits that are programmable allow children to understand ideas from computer science and engineering while simultaneously providing opportunities for expression through the creation of projects that can move around and respond to the environment through sensors. Despite the growing body of research demonstrating the benefits of using robotics with young children, there are very few robotics construction sets specifically designed for children in pre-kindergarten through second grade that offer both building and programming capabilities (Bers et al. 2013). In order to make technology and engineering instruction most useful for early childhood classrooms, there is a need for new technologies with design affordances and interfaces specifically developed for young learners (Bers et al. 2013).

The lack of developmentally appropriate robotics kits for young children led to the development of the newly developed KIWI technology used in this study. The KIWI prototype was developed by the DevTech Research Group at Tufts University, through funding from the National Science Foundation. KIWI involves hardware (the robot itself) and the software used to program KIWI called CHERP (Creative Hybrid environment for computer Programming). Over the past 3 years, KIWI has gone through several design iterations and has been tested in numerous public and private schools as well as in summer camp and lab settings (see Fig. 1). This testing was used to inform the re-design of KIWI to make it enjoyable and appealing to kids ages 4–7 while also sufficiently challenging.

Based on pilot testing and prior work with children, the criteria for the developmentally appropriate design of KIWI included the following: robotics parts should be physically and intuitively easy to sturdily connect, programming the robot should need the minimum amount of computer equipment, children should be able to attach a variety of crafts and recycled materials to the core robotic parts providing different types of creations, both stationary and mobile (Bers et al. 2013). The teachers' feedback that programming the robot should use minimal computer equipment, prompted researchers to create the version of KIWI used in this study which does not require any screen-time or even a computer to be programmed. This is additionally aligned with the American Academy of Pediatrics' recommendation that young children have a limited amount of screen time per day (<2 h) per day (American Academy of Pediatrics 2003; Barlow 2007).

Instead, its actions are programmed using CHERP, a tangible programming language consisting of interlocking wooden blocks (see Fig. 2). The robot has an embedded scanner

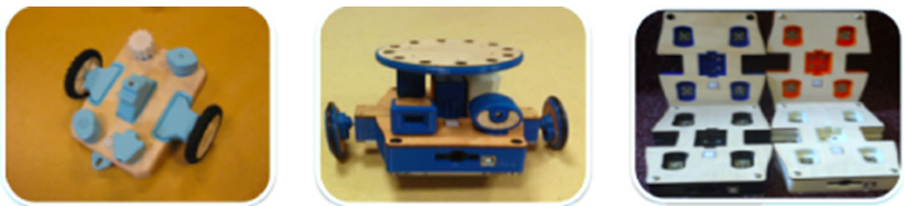


Fig. 1 Design iterations of KIWI prototype. The design changes the KIWI prototype went through based on pilot testing with children and teachers. Changes include: adding design spaces for building and creating with crafts, creating a see-through bottom to examine the inner workings of KIWI, and making the robots more colorful

Fig. 2 KIWI robot prototype and CHERP programming blocks

that allows users to scan barcodes on the CHERP blocks and send a program to the robot instantaneously.

Due to the limited number of robotics kits specifically designed for young children, much of the prior research on early childhood robotics uses commercially available materials designed for older children that have been adapted for use with younger children by the researchers. For example, prior research by Strawhacker and Bers (2014) used the commercially available LEGO® WeDO robotics kit (designed for children ages 7 and up) with Kindergarteners and adapted it for younger children by using a hybrid tangible-graphical software. While prior research by Bers et al. (2013) introduces the development of KIWI, the focus of that study was on teachers not children. This paper presents the first research study using KIWI with young children in order to determine what they can learn with the kit. Specifically, this paper investigates what children in pre-kindergarten through second grade learn about about robotics and programming concepts after completion of an 8 weeks curriculum using KIWI robotics and the CHERP programming language in their classrooms.

Methods

This classroom-based study collected data from children in pre-kindergarten through second grade upon their completion of an 8-week robotics curriculum using the KIWI robotics construction kit and CHERP programming language. Quantitative data was collected from the participating children in the form of two assessments: the Robot Parts test (which assessed robotics knowledge) and the Solve-Its tasks (which assessed programming knowledge).

Sample

A sample of N = 60 children in pre-kindergarten through 2nd grade participated in this research (N = 15 pre-kindergarteners, N = 18 kindergarteners, N = 16 first graders, N = 11 second graders) from an urban, public, early education school that serves children in Pre-K through third grade in Boston, Massachusetts. The school was selected as a research site in order to see how the robotics curriculum would unfold in a typical U.S. public school, outside of a research lab setting. The school selected was chosen to be as close as possible to a representative sample of public schools in the Boston area of Massachusetts. Massachusetts State demographic and census information reports that the school is 72 % Hispanic, 69 % Limited English Proficiency, 65 % Free or Reduced Lunch, 15 % Special Education (Massachusetts Department of Elementary and Secondary Education 2013). The classrooms selected represent a self-selected sample of teachers from this school that wanted to have robotics in their classroom.

Procedure

Over the course of 8 weeks, four classrooms (a pre-kindergarten, kindergarten, first grade, and second grade classroom) completed an introductory robotics and programming curriculum taught by research assistants from the DevTech Research Group at Tufts University. Lessons were taught once a week and lasted approximately 1 h. The children's regular classroom teachers were in the classroom to assist with behavioral management. A larger goal for the robotics work at this site was to have classroom teachers learn by observing trained research assistants so that they would be able to implement the robotics curriculum without outside help in the future. In order to gain the self-confidence to do this, the classroom teachers occasionally assisted with the instruction of robotics concepts in this study.

Curriculum

The robotics lessons were integrated with a larger curriculum unit called "Me and My Community" in which students explore their own identities, their school community, and their neighborhoods. The robotics component of each lesson involved a hands-on building or programming task that children completed in groups of two to three. Each lesson was building on concepts taught the previous week, leading up to a culminating project with an interactive robot map representing the community. In the following sections, more details will be provided regarding the curriculum.

Lessons 1 and 2: What is a robot and what is programming?

During the first two lessons of the curriculum, children explored the basic robotic parts of the KIWI robot (motors, sensor modules and art platforms) and learned how to sequence a program using the CHERP wooden blocks. Children used robotic parts as well as arts and recycled materials to create a sturdy "Self Portrait Robot" designed to represent themselves (see Fig. 3). Next, they used the CHERP blocks to program their robots to dance the Hokey-Pokey. After mastering this basic sequencing exercise, children programed their robots to dance to music of their choice from around the world. While the kindergarten through second grade classes finished these tasks within the first 2 weeks, the pre-kindergarten classes spent the majority of the 8 weeks reinforcing concepts about programming sequence and sturdy building.

Fig. 3 Child-made self-portrait robot

Lesson 3: What is a sound sensor?

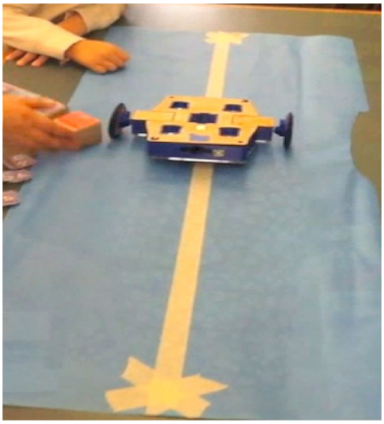
After children mastered basic robotic and programming concepts, they were introduced to the concept of sensors. They started by going on a "sensor walk" around the school trying to find all automatic sensors in their building. Next, children incorporated a sound sensor into their dance program from the previous lesson. Most programs involved a robot following a sequence of dance movements (forward, backward, shake, etc.) and then, once the sound sensor detects a clap, the robot performs a different actions or makes a sound. For example, a sequence of blocks might read as follows: turn left, shake, wait for clap, spin, sing, end (see Fig. 4). The robot would begin the dance by turning left and shaking, but then stop and remain motionless until the sound sensor hears the child clap. Then, it will continue the dance by spinning and playing a singing noise.

Lesson 4: What are repeat loops?

Next, students went on to learning how to make a segment of their code repeat an action a particular number of times using Repeat Loops. On a basic map on the floor, children programmed their robots to go from point A to point B using number parameters (see Figs. 5, 6). Children could create a Repeat Loops program such as: Begin, Repeat three times, Forward, End Repeat, End. Children would have to test out different number parameters to figure out how many times the forward command needs to repeat in order to get the robot across the map and stop at the appropriate spot.

Fig. 4 Sample program for the sound sensor

Fig. 5 Repeat Loops floor map. This figure illustrates maps children created for their robots to navigate from one X to another using Repeat Loops and Number Parameters



Lesson 5: What are distance and light sensors?

Using the same floor map from the previous lesson, children used distance and light sensors to have their robots start at point A and stop at point B. Instead of using number parameters to tell their robots how many times to move forward, they used sensor parameters so that the robot would only stop after input from a sensor was triggered (see Figs. 7, 8).

Lesson 6: What are conditional statements?

After children had mastered the concept of Repeat Loops and sensors, they used conditional "If" branches in their programs to have their robot carry out different actions depending on sensor input. Due to the amount of time spent mastering previous concepts, only the second grade class was introduced to the concept of conditional statements. In this lesson, children in second grade created programs so that their robots would do different things depending on if was "day" or "night" (i.e. "Bright" or "Dark") according to light

Fig. 6 Repeat Loops Program. This figure illustrates a CHERP program that would tell the KIWI robot to move forward three times and then stop

Fig. 7 Child uses flashlight to trigger light sensor

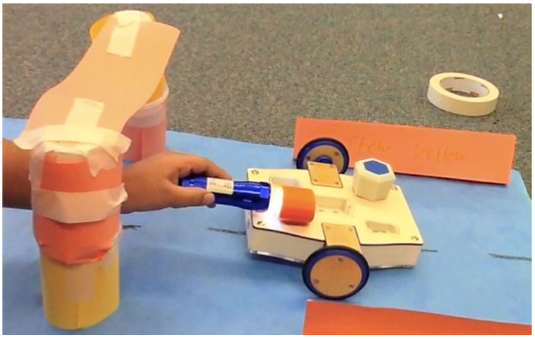


Fig. 8 Programming blocks for Light Sensor Program. Using this type of program, children programmed their robots to continue forward along the map until a flashlight shines into the light sensor. Once the light is detected, the robot stops moving

Fig. 9 Conditional Statement Program. Children created conditional programs that allowed their robots to sing if it was "daytime" (i.e. the light sensor detects brightness) and "go home" (i.e. move forward) if it is "not daytime" (i.e. the light sensor does not detect brightness)

sensor input. For example, one child created a program that would let their robot sing if it was "daytime" (i.e. robot senses it is bright). However, if it were "night" (i.e. robot senses it is *not* bright) the robot would go straight home (see Fig. 9).

Lesson 7: final project

As a culminating project, children from all of the grades created floor maps of their neighborhood including their homes, school, and other landmarks of significance to them. The children programmed their robots to drive along the map stopping at significant locations and carrying out different actions once there (see Fig. 10). For the pre-kindergarten class, maps and programs were relatively simple. The projects got increasingly complex for the older children. For example, the second graders used sensors to have their robots move along the map and stop at more than one location in the neighborhood.

Pace of curriculum

While all of the grades participated in 8 weeks of robotics and programming, each classroom moved at a pace that was comfortable for the students and thus, not all grades covered all of the topics. The pre-kindergarten class spent the 8 weeks familiarizing themselves with basic robotics concepts, sturdy building, and figuring out how to sequence a basic program in a syntactically correct way. Meanwhile, the first and second graders moved past these foundational concepts quickly and were able to spend time experimenting with the different sensors, and using repeat loops or conditional statements to create complex programs for their robots.

Fig. 10 Child-made KIWI robot programmed to drive along a neighborhood map

Assessments

After curriculum implementation was complete, two assessments were used to measure children's robotics and programming knowledge. The Robot Parts task was used to determine each child's robotics knowledge and the Solve-It assessment was used to measure each child's programming knowledge. Each of these assessments was completed on an individual basis. Because children worked in small groups during the curricular activities, it was important to implement assessments as individual tasks to see what types of challenges children could solve on their own.

Robot parts task

The "Robot Parts" task was newly developed by the DevTech Research Group for this study in order to assess children's individual knowledge of the different KIWI robotic parts and their functions. Children sat down with a researcher one on one and were asked to help the researcher identify different parts of the robot and their functions. For example, researchers asked, "what part should I use if I want my robot to turn its light on?" with the correct answer being the light output. After each question, children would pick up or point to the part they thought was correct and their answers were recorded. Children were asked to identify five different robot parts: three sensors (light, distance, and sound), the light output, and the motors. Children received a point for correctly identifying each of these five components, yielding scores that could range from 0 (could not identify any of the parts) to 5 (correctly identified all parts).

Solve-Its

The "Solve It" tasks were used to assess children's individual programming knowledge at the end of curriculum implementation. The Solve-It tasks were developed to target areas of foundational programming ability and basic sequencing skills (Strawhacker et al. 2013). This assessment captures student mastery of programming concepts, from basic sequencing up through conditional branching.

The assessments were verbally administered to each class by the same researcher, with the assistance of the regular classroom teacher. This assessment required children to listen as the researcher told them a series stories about a robot. Then, working individually, children attempted to create the robot's program on a piece of paper using the CHERP programming icons provided to them (see Fig. 11). For example, one story was about the bus from the children's song "Wheels on the Bus". Children already knew the song and had a reference point for how that robot (the bus) should behave.

Eight Solve-Its were administered to children upon completion of the curriculum. The eight Solve Its tested the following concepts: Easy Sequencing, Hard Sequencing, Easy Repeat Loops with Number Parameters, Advanced Repeat Loops with Number Parameters, Easy Sequencing with the "Wait-For" Command, Easy Repeat Loops with Sensor Parameters, Hard Repeat Loops with Sensor Parameters, Conditional Branching. Tasks were called "easy" or "hard" based on how many commands children needed to sequence (i.e. easy tasks had fewer blocks for children to sequence than hard tasks).

Each of the eight Solve-It tasks described above was scored on a 0–6 rubric based on how close the child's program came to being completely correct (a score of 6). The scoring rubric was developed and piloted by the DevTech Research Group (Strawhacker et al.

Fig. 11 Sample child-completed Solve-Its

2013; Strawhacker and Bers 2014). Each question received a total score based on separate criteria, including placement of begin and end blocks and relative order of action blocks. The scoring rubric was developed after a pilot assessment was administered, to identify incorrect answer patterns that could demonstrate developmental level rather than programming comprehension. Inter-scorer reliability tests showed precise agreement (two items; K = 0.902, $p < 0.001$) (Strawhacker and Bers 2014). The score of 0–6 was derived from sub-scores targeting concepts of control flow and actions equencing. For the purposes of this paper, only the total scores are considered for analysis.

Results

For all tasks on the Robot Parts Test and the Solve-Its assessment, basic descriptive statistics were calculated as well as statistical comparisons between the four grades. On average, the children in this study were highly successful in mastering basic programming and robotics concepts after completing this curriculum. Performance typically increased by grade level, though not for every task. Detailed analysis is presented in the following sections.

Robotics knowledge

Children in kindergarten through second grade completed the Robot Parts post- test upon completion of the curriculum. Children could receive scores ranging from 0 to 5 depending on whether or not they correctly answered each of the five questions on the assessment. Average total scores on the Robot Parts test were very high for all grades, indicating that overall, children had a good understanding of the functions of each part of the KIWI robot. Mean scores on the test increased as the children got older.

The Kruskal–Wallis H test was used to determine if there were significant differences on the Robot Parts task performance based on the child's grade. The Kruskal–Wallis test was chosen over an ANOVA because the data did not meet all of the criteria to perform parametric testing, most likely due to the small sample sizes for each of the classes. No significant differences were found based on the child's grade, indicating that all children were able to master the KIWI robotics concepts similarly regardless of what grade they were in.

Children did very well identifying the motors and the lantern (light output) and had more difficulty identifying each of the sensors (see Fig. 12). In particular, the light sensor

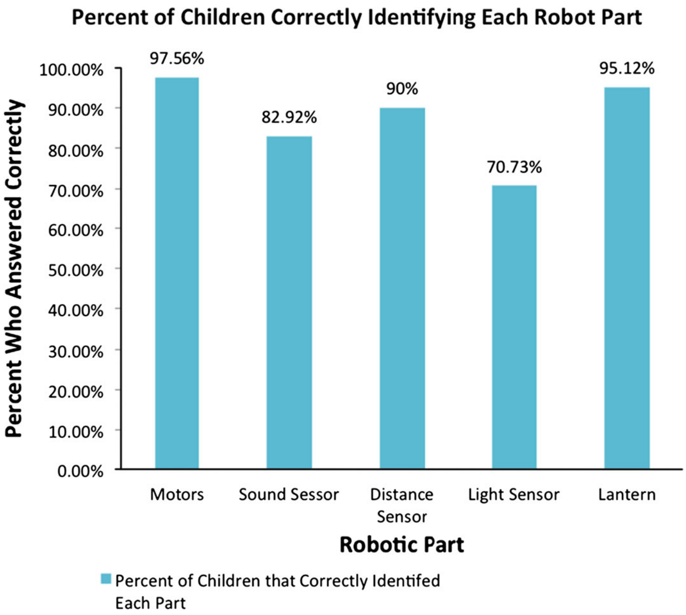


Fig. 12 Percent of children able to identify robotic parts correctly

was difficult for some children to identify (70.73 % were able to correctly identify it), even after completing lessons that used the light sensor. It is possible that the lantern and the light sensor were too similar in their design and function for children to differentiate the concept of input and output (one takes *in* light while the other puts *out* light).

Programming knowledge

Children in pre-kindergarten through second grade completed various Solve-It programming tasks upon completion of 8-week curriculum. Which Solve-Its they completed depended on the grade they were in. Children were not given a task that addressed a programming concept that was not covered in their curriculum.

Mean scores show that across all grades, children were highly successful in their performance on the Solve-It tasks. Each Solve-It was scored on a scale from 0 to 6, and all grades had a mean score above 3 on all of the tasks administered (see Table 1). On almost all of the Solve-Its, the second grade class had the highest mean scores. The kindergarten, first, and second graders performed better on the sequencing Solve-It tasks (Solve-Its 1, 2, and 5) than the Repeat Loop and Conditional Statement tasks (Solve-Its 3, 4, 6, 7, and 8). Children made a variety of different types of mistakes on the Solve-Its, some of which were syntactical (i.e. the program they created had a logical programming error and would not be able to work on a real robot) and some of which were story related (i.e. the program made syntactical sense, however, did not match the sequence of the story).

Pre-kindergarten only completed the Sequencing Solve-Its because they did not cover the advanced programming concepts in their curriculum. The pre-kindergarten class performed better on Solve-It 1 (the easy sequencing task) which only asked them to sequence four instructions correctly. They did not score as well on Solve-It 2 (the hard sequencing

Table 1 Mean Solve-Its scores by grade

	Solve-It 1	Solve-It 2	Solve-It 3	Solve-It 4	Solve-It 5	Solve-It 6	Solve-It 7	Solve-It 8
Pre-K	5.2 (1.521)	3.4 (1.957)	N/A	N/A	N/A	N/A	N/A	N/A
K	5.29 (1.437)	5.5 (1.019)	4.57 (1.222)	4.57 (1.089)	6 (0.000)	3.71 (1.069)	4.36 (1.008)	N/A
1st	6 (0.000)	5.19 (1.276)	4.94 (1.124)	4.5 (.632)	5.63 (1.025)	3.69 (.793)	4.13 (.719)	N/A
2nd	6 (0.000)	6 (0.000)	4.91 (1.044)	5 (1.000)	6 (0.000)	4.55 (1.214)	4.45 (1.128)	4.9 (1.197)

Not all grades completed all Solve-Its. The Solve-Its that were administered depended on the concepts that were covered in their robotics curriculum. Mean scores with standard deviations in parentheses. Solve-It numbers correspond to the following concepts: Solve-It 1: Easy Sequencing, Solve-It 2: Hard Sequencing, Solve-It 3: Easy Repeat Loops with Number Parameter, Solve-It 4: Hard Repeat Loops with Number Parameter, Solve-It 5: Easy Sequencing Wait-For Command, Solve-It 6: Easy Repeat Loops with Sensor Parameter, Solve-It 7: Hard Repeat Loops with Sensor Parameter, Solve-It 8: Easy Conditional Statement with Sensor Parameter task), which asked them to sequence five instructions correctly. There are many reasons why the pre-kindergarten class may have done better on the easy sequencing task than on the hard sequencing task that are not necessarily related to programming knowledge. For example, the easy sequencing task had fewer instructions and this may have simply been more appropriate for the students' working memory and attention span.

The Kruskal–Wallis H test was used to determine if there were significant differences on Solve It performance based on the child's grade. The Kruskal–Wallis test was chosen over an ANOVA because the data did not meet all of the criteria to perform parametric testing, most likely due to the small sample sizes for each of the classes. The Kruskal–Wallis H test showed that there was a statistically significant difference in Solve Its score between the different grades for the Hard Sequencing Solve It, $\chi^2(3) = 19.33$, $p = .00023$, with a mean rank score of 16 for pre-kindergarten, 32.86 for kindergarten, 29.88 for first grade, and 38 for second grade. There were no other significant differences based on grade for any of the other Solve Its.

Because the Kruskal–Wallis H test is an omnibus test statistic, post hoc testing was used to determine which grades were significantly different from one another for the Hard Sequencing task. Kindergarten ($U = 40.5$, $p = .002$), first ($U = 57.5$, $p = .012$), and second grade ($U = 22$, $p = .001$), each performed statistically significantly better than the pre-kindergarten class on Hard Sequencing. There were no other statistically significant differences between the grades on this Solve-It task. This indicates that the kindergarten, first, and second grade classes performed equally well on the advanced programming Solve-Its.

Discussion

Using the KIWI robotics kit and the CHERP programming blocks, even the youngest children in this study (the pre-kindergarten class) were able to successfully master sequencing a syntactically correct program for their robots. However, having more instructions to sequence was more challenging for the pre-kindergarten students than sequencing a shorter program, even though the programming concepts utilized were the same in both tasks. This may be due to their working memory and the capacity to remember all the parts of a longer story at any given time. Working memory is described as a system for holding and manipulating information over brief periods of time (Abreu et al. 2010). Working memory can be assessed by tasks that involve the simultaneous processing and storage of information (Daneman and Carpenter 1980). In the case of the Solve-Its, children were asked to simultaneously process a story being told, remember the programming instructions they had learned over the course of 8 weeks, *and* connect the instructions to the story. All of these simultaneous tasks may have been too heavy a load for the youngest children in this study, even if the programming concepts were manageable. Future work may want to look at alternative ways to assess children in pre-kindergarten.

Each of the grades (kindergarten, first, and second) performed significantly better on the Hard Sequencing task than the pre-kindergarten class. With the exception of this task, there were no other significant differences on Solve-Its performance between the grades. Since the pre-kindergarteners did not complete the other Solve-Its, this indicates that kindergarteners, first graders, and second graders were all able to master skills related to sequencing, repeat loops with number and sensor parameters, and the wait-for command equally well.

When it came to mastery of the sensor modules and motor components of the KIWI robot, all of the children who completed this assessment scored very highly overall, indicating that the kit was appropriate for the children to work with. Additionally, it indicates that the curriculum was generally successful at teaching the participating children how to identify the look and functions of various sensors, output, and motors. The robotic part that was the most difficult for the children to identify was the light sensor. This is not surprising considering the light sensor and the light output were often confused by children during curricular activities as well. Future work will need to re-consider the design of the light sensor and the lantern to determine if making these two parts look more distinct from one another will help children differentiate their functions. Future work may also want to explore alternative ways to assess young children's understanding of the different robot parts. For example, instead of asking children to answer a question about the robotic parts, children could be asked to complete a hands-on task with the robot.

Although there were very few differences in performance on both the programming and robotics post-tests based on the children's grade, there were differences in how the children progressed through the curriculum. The second graders were able to progress through the material more quickly and gain mastery of these concepts with less instruction and help. They were the only class that progressed through the curriculum fast enough to begin covering conditional branching in their programs by week 8. Had the curriculum continued, they may have been ready to progress from this simple robotics kit and move onto a commercially available robotic kit for older children such as the LEGO® WeDO robotics construction set.

Meanwhile, the pre-kindergarten class progressed through the curriculum very slowly, and the focus of the classes became working on understanding the correspondence between the CHERP programming blocks chosen and robot's actions. This exploration was continued with repetition over the course of the 8 weeks in order to allow the children to walk away with a mastery of the basic programming commands. The pre-kindergarten students also took more time becoming comfortable scanning and handling the robot (although, they were all able to scan programs onto their robots themselves by the end of the 8-week curriculum). This falls in line with prior research that found that pre-kindergarten children spend more time on basic robotics concepts than children in kindergarten and older grades and move through introductory robotics curriculum at a slower pace (Sullivan et al. 2013). Prior work with pre-kindergarten students and robotics has also discussed the importance of one-on-one adult assistance during building and programming activities (Sullivan et al. 2013). It is possible that with more adults, the pre-school children in this study may have been able to complete more of the lessons in the curriculum, such as sensors or repeats.

Limitations and future work

This study collected limited data from the participating pre-kindergarten children. The 8-week curriculum moved at the pace of each class, and this time allotment was too short to allow the pre-kindergarteners to learn more advanced programming concepts, and thus, they were only able to complete two Solve-It tasks. Additionally, the pre- kindergarten class struggled with mastery of the robotic parts and was not introduced to the KIWI sensors at all in their curriculum. Therefore, they did not complete the Robot Parts Task as a post-test. Because of this, there was limited data that could be used from the pre-kindergarteners to make grade comparisons in this study.

The pre-kindergarten children also worked at a slower pace than the older children. Although they were able to decorate and build their robots for their final map projects they were not able to use sophisticated programming concepts, only basic ones. More research is needed to understand whether the pre-kindergarten students could have mastered other concepts if they were given more time or if robotics were part of their curriculum year round. The results from this study fall in line with prior research that has highlighted the challenges that arise in pre-kindergarten classes and the need of adult support and different strategies working with children this young (Sullivan et al. 2013).

The data analyzed in this study was purely quantitative and looked only at mean performance by grade level. It is beyond the scope of this study to discuss more nuanced details such as the problem solving strategies children employed on the different tasks or the styles of mistakes that children made on the programming tasks. Future research should look at the qualitative differences in children's programming ability and approaches to robotics and programming tasks. These types of details could allow teachers to create curricular activities and assessments that are appropriately challenging and engaging for children in pre-kindergarten through second grade.

Another limitation of this study is that the robotics curriculum was taught by research assistants and not by regular classroom teachers. It is possible that it was not the tools and curriculum alone that led to students' mastery of the concepts taught, but having these experts in the classroom may have contributed as well. Future research will follow classroom teachers using innovative new tools such as the KIWI robotics kit in order to ensure that they are not only developmentally appropriate for children to use, but easy for early childhood instructors to teach with as well.

Finally, this study is limited by the small sample size that prevented the use of parametric statistical testing (non-parametric tests were used here). This study was a pilot and future work with larger sample sizes will be conducted in order to gather more widely generalizable results. Results from this and future work will inform the re-design of the KIWI robotics kit and accompanying curriculum.

Conclusion

Robotics and programming offers early childhood teachers a new and exciting way to address the "T" of technology and "E" of engineering that are most neglected in early childhood STEM education. This study demonstrates that in as early as pre- kindergarten, children are able to master foundational concepts regarding programming a robot and that children as young as 7 years old are able to master concepts as complex as programming a robot using conditional statements. The KIWI robotics prototype used in this study offered the added advantage of allowing children to explore these concepts without the use of any computers or screen time.

This study provides preliminary evidence that robotics construction kits specifically designed for young learners can be a useful and educational tool in early childhood classrooms. Robotics is not only an engaging activity, but it can also be integrated into other curricular units happening in the classroom (such as the "Me and My Community" project described here). Children not only have fun working with robotics, but research shows positive learning outcomes as well.

References

Abreu, P., Conway, A., & Gathercole, S. (2010). Working memory and fluid intelligence in young children. *Intelligence, 38*(2010), 552–561.

American Academy of Pediatrics. (2003). Prevention of pediatric overweight and obesity: Policy statement. *Pediatrics, 112*, 424–430.

Barlow, S. E., & the Expert Committee. (2007). Expert committee recommendations regarding the prevention, assessment, and treatment of child and adolescent overweight and obesity: Summary report. *Pediatrics, 120*, S164–S192.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads, 2*(1), 48–54.

Barron, B., Cayton-Hodges, G., Bofferding, L., Copple, C., Darling-Hammond, L., & Levine, M. (2011). *Take a giant step: A blueprint for teaching children in a digital age*. New York: The Joan Ganz Cooney Center at Sesame Workshop.

Bers, M. (2008). *Blocks to robots: Learning with technology in the early childhood classroom*. New York: Teachers College Press.

Bers, M., & Horn, M. (2010). Tangible programming in early childhood: Revisiting developmental assumptions through new technologies. In I. R. Berson & M. J. Berson (Eds.), *High-tech tots: Childhood in a digital world* (pp. 49–70). Greenwich: Information Age Publishing.

Bers, M., Ponte, I., Juelich, K., Viera, A., & Schenker, J. (2002). Teachers as designers: Integrating robotics in early childhood education. *Information Technology in Childhood Education* AACE, pp. 123–145.

Bers, M. U., Seddighin, S., & Sullivan, A. (2013). Ready for robotics: Bringing together the T and E of STEM in early childhood teacher education. *Journal of Technology and Teacher Education, 21*(3), 355–377.

Brosterman, N. (1997). *Inventing kindergarten*. New York: H.N. Abrams.

Cejka, E., Rogers, C., & Portsmore, M. (2006). Kindergarten robotics: using robotics to motivate math, science, and engineering literacy in elementary school. *International Journal of Engineering Education, 22*(4), 711–722.

Clements, D. H. (1999). Young children and technology. In G. D. Nelson (Ed.), *Dialogue on early childhood science, mathematics, and technology education*. Washington, DC: American Association for the Advancement of Science.

Daneman, M., & Carpenter, P. A. (1980). Individual-differences in working memory and reading. *Journal of Verbal Learning and Verbal Behavior, 19*(4), 450–466.

International Society for Technology in Education. (2007). NETS for students 2007 profiles. Washington, DC: ISTE. Retrieved from www.iste.org/standards/nets-for-students/nets-for-students-2007-profiles.aspx#PK-2

International Society for Technology in Education and The Computer Science Teachers Association. (2011). *Operational definition of computational thinking for K-12 thinking*. International Society for Technology in Education and The Computer Science Teachers Association.

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., et al. (2011). Computational thinking for youth in practice. *ACM Inroads, 2*(1), 32–37.

Lee, K., Sullivan, A., & Bers, M. U. (2013). Collaboration by design: Using robotics to foster social interaction in kindergarten. *Computers in the Schools, 30*(3), 271–281.

Madill, H., Campbell, R. G., Cullen, D. M., Armour, M. A., Einsiedel, A. A., Ciccocioppo, A. L., et al. (2007). Developing career commitment in STEM-related fields: myth versus reality. In R. Burke, M. Mattis, & E. Elgar (Eds.), *Women and minorities in science, technology, engineering and mathematics: Upping the numbers* (pp. 210–244). Northhampton, MA: Edward Elgar Publishing.

Markert, L. R. (1996). Gender related to success in science and technology. *The Journal of Technology Studies*, *22*(2), 21–29.

Massachusetts Department of Elementary and Secondary Education (MA DOE). (2013). *Enrollment data.* Malden, MA: Massachusetts Department of Elementary and Secondary Education. Retrieved from http://profiles.doe.mass.edu/profiles/student.aspx?orgcode=00350009&orgtypecode=6&

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

NAEYC & Fred Rogers Center for Early Learning and Children's Media. (2012). Technology and interactive media as tools in early childhood programs serving children from birth through age 8." Joint position statement. Washington, DC: NAEYC; Latrobe, PA: Fred Rogers Center for Early Learning at

Saint Vincent College. Retrieved from www.naeyc.org/files/naeyc/file/positions/PS_technology_WEB2.pdf

Perlman, R. (1976). *Using computer technology to provide a creative learning environment for preschool children*. Logo memo No. 24, Cambridge, MA: MIT Artificial Intelligence Laboratory Publications, 260 pp.

Resnick, M., Martin, F., Berg, R., Borovoy, R., Colella, V., Kramer, K. et al. (1998). Digital manipulatives. *Proceedings of the CHI '98 conference*, Los Angeles, April 1998.

Resnick, M. (2013). Learn to Code, Code to Learn. EdSurge, May 2013.

Sesame Workshop. (2009). Sesame workshop and the PNC Foundation join White House effort on STEM education. Retrieved from http://www.sesameworkshop.org/newsandevents/pressreleases/stemeducation_11212009

Siu, K., & Lam, M. (2003). Technology education in Hong Kong: International implications for implementing the "Eight Cs" in the early childhood curriculum. *Early Childhood Education Journal, 31*(2), 143–150.

Steele, C. M. (1997). A threat in the air: How stereotypes shape intellectual identity and performance. *American Psychologist, 52*, 613–629.

Strawhacker & Bers (2014). "I want my robot to look for food": Comparing children's programming comprehension using tangible, graphical, and hybrid user interfaces. *International Journal of Technology and Design Education.* Advance online publication. doi: 10.1007/s10798-014-9287-7

Strawhacker, A., Sullivan, A., & Bers, M. U. (2013). TUI, GUI, HUI: Is a bimodal interface truly worth the sum of its parts? *Proceedings of the 12th international conference on interaction design and children (IDC '13)* (pp. 309–312). New York: ACM.

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, pp. 203–219. Retrieved from http://www.jite.org/documents/Vol12/JITEv12IIPp203-219Sullivan1257.pdf

U.K. Department for Education. (2013). *The national curriculum in England curriculum framework document*.

U.S. Department of Education, Office of Educational Technology. (2010). *Transforming American education: Learning powered by technolog*y. Washington, DC. Retrieved from http://www.ed.gov/technology/netp-2010

White House. (2011). *Educate to innovate*. Retrieved from: http://www.whitehouse.gov/issues/education/educate-innovate

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

Wyeth, P. (2008). How young children learn to program with sensor, action, and logic blocks. *International Journal of the Learning Sciences, 17*(4), 517–550.