

Imagining, Playing, and Coding with KIBO: Using Robotics to Foster Computational Thinking in Young Children

Amanda A. Sullivan^{1*}, Marina Umaschi Bers², Claudia Mihm³
The DevTech Research Group^{1 2 3}
Tufts University, United States^{1 2 3}

Amanda.Sullivan@tufts.edu, Marina.Bers@tufts.edu, Claudia.Mihm@tufts.edu

ABSTRACT

The KIBO robotics kit offers a playful and tangible way for young children to learn computational thinking skills by building and programming a robot. KIBO is specifically designed for children ages 4-7 years old and was developed by the DevTech research group at Tufts University through nearly a decade of research funded by the National Science Foundation. KIBO allows young children to become engineers by constructing robots using motors, sensors, and craft materials. Children also become programmers by exploring sequences, loops, and variables. Through programming KIBO, children engage with computational thinking skills and ideas including algorithms, modularity, and control structures. Unlike other programming interfaces for children, the KIBO robot is programmed to move or to respond to sensor input by using tangible programming blocks—no computer, tablet, or screen-time required. This paper provides an overview of the design features of KIBO and a synthesis of the research that has been done throughout the development of this kit. It provides examples of curriculum for playfully engaging young children with computational thinking using KIBO.

KEYWORDS

Early childhood, engineering, robotics, programming, computational thinking

1. INTRODUCTION

Early childhood is an important time for young children to grow, play, and explore the world they live in. Developmentally, it is a life stage characterized by genuine curiosity and desire for learning. In order for young children to master new knowledge about the world, they need hands-on experiences to construct their learning (Piaget, 1936). New technologies such as robotics kits and coding applications offer children a hands-on way to learn about many of the things they encounter every day but do not understand, such as sensors, batteries, and lights (Papert, 1980). Robotics is an ideal tool for early childhood because it facilitates cognitive as well as fine motor and social development (Bers, 2008; Clements, 1999; Lee, Sullivan, & Bers, 2013; Svensson, 2000). It engages children creatively, as an expressive medium, allowing young children to 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, 2017; Bers 2008).

When learning to build and program robots, young children are also engaging in a type of problem solving and analysis called *computational thinking*. The term “computational thinking” can be defined as solving problems algorithmically and developing a sense of technological fluency (Bers, 2017; Bers, 2010; Papert, 1980) Children as young as four years old can learn foundational computational thinking concepts (Bers, 2017; Bers, 2008) and this kind of learning can support their literacy, mathematical, and socio-emotional development (Kazakoff & Bers, 2012; Kazakoff, Sullivan, & Bers, 2013). While computational thinking is rooted in computer science, many have argued that it is a universally applicable attitude and skillset that is fundamental for everyone to master, just like reading, writing, and arithmetic (Wing, 2006).

KIBO (see Figure 1) was born out of research led by Marina Bers at the DevTech Research Group at Tufts University (Bers, 2017). The goal was to foster playful exploration of computational thinking during early childhood through tangible objects. Later on, KIBO became commercially available through KinderLab Robotics with funding from the National Science Foundation and a successful Kickstarter campaign (Bers, 2017). KIBO’s design was based on years of child development research in collaboration with teachers and early childhood experts to meet the learning needs of young children in a developmentally appropriate way (Sullivan, Elkin, & Bers, 2015; Sullivan & Bers, 2015; Kazakoff & Bers, 2014). This paper provides an introduction to the design of KIBO and presents an overview of the worldwide research conducted with KIBO for the last several years to promote computational thinking in young children.



Figure 1. KIBO robot with a sample block program, art platforms, and art supplies for decorating.

2. DESIGN FEATURES OF KIBO

KIBO is a robotics construction kit that involves both hardware (the robot itself) and software (tangible programming blocks) used to make the robot move. The kit contains easy to connect construction materials including: wheels, motors, light output, and sensors as well as a variety of art platforms (See Figure 1 on the previous page).

KIBO is programmed using interlocking wooden programming blocks (see Figure 2). These wooden blocks contain no embedded electronics or digital components. Each wooden block has a colorful label with an icon, text and a bar code; as well as a hole on an end and a peg on the other. The KIBO robot has an embedded scanner that allows users to scan the barcodes on the programming blocks and send a program to their robot instantaneously. No computer, tablet, or other form of “screen-time” is required to learn programming with KIBO. This is aligned with the American Academy of Pediatrics’ recommendation that young children have a limited amount of screen time per day per day (American Academy of Pediatrics, 2016).



Figure 2. KIBO’s tangible programming language. Each block has a unique barcode that is scanned by the robot.

This programming language was inspired by early ideas from tangible programming beginning with Radia Perlman’s work in the mid 1970’s (Perlman, 1976) and revived by the work of Suzuki & Kato (1995) nearly two decades later. In recent years, there have been several tangible languages have been created in a number of different research labs around the world (e.g. McNerney, 2004; Smith, 2007; Horn & Jacob, 2007).

In contrast to graphical programming, which relies on pictures and words on a computer screen, tangible programming uses physical objects to represent the same concepts (Manches & Price, 2011). Wooden programming blocks are naturally familiar and comfortable for children, in the tradition of learning manipulatives already used in early childhood classrooms to teach shapes, size, and colors (Froebel, 1826; Montessori & Gutek 2004). KIBO’s programming blocks are shared easily and manipulated by young users with limited fine motor capacity.

KIBO’s programming language is composed of over 18 individual wooden programming blocks. Some of these

blocks represent simple motions for the KIBO robot such as, move Forward, Backward, Spin, and Shake. Other blocks represent complex programming concepts such as Repeat Loops and Conditional “If” statements that involve sensor input (See Figure 3).



Figure 3. This figure provides an example of a conditional statement with KIBO.

3. EARLY COMPUTATIONAL THINKING

3.1. What is Computational Thinking?

In recent years, there has been a growing focus on improving children’s technological literacy and making computational thinking a priority in early childhood school settings in the United States (e.g. U.S. Department of Education, 2010). According to Wing (2006) computational thinking is defined as, “solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science” (p.33). *Computational thinking* involves a set of skills that include problem-solving, design and expression, and systematic analysis (Bers, 2017; Bers, 2010). Computational thinking represents a type of analytical thinking that shares many similarities with mathematical thinking (e.g., problem solving), engineering thinking (designing and evaluating processes), and scientific thinking (systematic analysis) (Bers, 2017).

Brennan & Resnick (2013) explain that computational thinking involves the *concepts* designers engage with as they program, the *practices* designers develop as they engage with the concepts, and the *perspectives* designers form about the world around them and about themselves. Concepts may include very specific programming concepts (such as repeat loops or conditional statements), the practices may include methods of problem-solving or collaboration, and perspectives may include questioning things beyond the interface you are working with (such as questioning how other things in the world are automated, besides KIBO). Bers (2017) expands on the notion of computational thinking, describing it not only as a problem solving process, but as an *expressive process*; a skillset that allows for new ways to communicate, to tell stories and convey ideas.

It is important to note that there are many non-technical and even non-academic examples of instances that call for computational thinking skills (Wing, 2008; Yadav, 2011). These everyday activities draw on the same type of problem solving, but do not involve programming. Wing (2008) presents a series of examples including: sorting Legos (using the concept of “hashing” to sort by color, shape, and size), learning to cook a meal (using “parallel

processing” to manage cooking at different temperatures for different amounts of time) and looking up your name in an alphabetical list (linear: starting at beginning of the list, binary: starting at the middle of the list). Each of these examples are activities young children are beginning to encounter in their everyday lives.

3.2. *Fostering Computational Thinking with KIBO*

KIBO is designed to promote a specific set of computational thinking skills. KIBO aims to foster seven “powerful ideas” of computational thinking described by Bers (2017). These ideas include: 1) algorithms, 2) modularity, 3) control structures, 4) representation, 5) hardware/software, 6) the design process, and 7) debugging. Table 1 below describes these concepts and how children explore them with KIBO. In the following section we provide examples of curricular units that foster these computational thinking concepts in a hands-on and playful way.

Table 1. Computational Thinking Concepts Explored with the KIBO Robotics Kit

Concept	Examples
Algorithms	Children use KIBO to explore logical organization and sequencing using the tangible programming blocks
Modularity	Children learn how to break up a large job into smaller steps when programming KIBO to navigate mazes or complete challenges
Control Structures	Children explore the ways KIBO can make decisions based on conditions using Repeat Loops and Conditional Statement blocks
Representation	Children learn that the colors and symbols on the blocks represent different types of actions
Hardware & Software	Children learn that computing systems, like KIBO, need both hardware (robotic parts) and software (blocks) to operate
Design Process	Children move through an iterative process used to develop programs and tangible artifacts
Debugging	Children troubleshoot their code when KIBO does not behave as expected

4. KIBO CURRICULUM

While the act of coding often evokes a very serious image of someone quietly working through lines of code on a computer, KIBO offers a more playful approach that is aligned with the spirit of early childhood education. Play in early childhood is not just fun; research has shown that it enhances children’s capacity for cognitive flexibility and, ultimately, creativity (Russ, 2004; Singer & Singer, 2005).

The DevTech Research Group has developed over a dozen curriculum units that focus on playful learning with KIBO in order to teach the computational thinking skills listed in the previous section. These curricular units also focus on STEAM (Science, Technology, Engineering, Arts, and Mathematics) content integration. In this section, we provide three examples of STEAM curriculum designed for KIBO: *Dances from Around the World*, *Robotic Animals*, and *Patterns All Around*. These illustrate how KIBO can be used to explore computational thinking while teaching other STEAM content such as dance, social studies, and math.

4.1. *Dances from Around the World*

The *Dances from Around the World* (DevTech, 2015) unit is designed to combine music, culture, dance, and language with programming and engineering content. The end project involves children programming their KIBOs to perform their favorite dance from anywhere in the world. It is completed over the course of approximately seven weeks. Each week, teachers introduce new robotics and programming concepts, from basic sequencing through conditional statements, to their students within the curriculum’s music and dance theme. For the final project, students work in pairs or small groups to design, build, and program a dance of their choosing. This involves not only robotics and programming knowledge, but also research into the music, history and cultural relevance of the dance, and facts about the country or culture in which the dance originated. The unit culminates in a dance recital for both the children and the robots to perform in together. Children engaged with open-ended free-play time to listen to their chosen music and come up with a dance on their own.

While this project engaged children with all of the seven powerful ideas of computational thinking described in Table 1, children had to devote particular focus on the idea of **sequencing** when choreographing and programming their robot dances. They had to carefully consider the timing of the music and any traditional dance steps that needed to be included (and if so, in what order). They needed to program their robot’s actions in a sequential order that matched the order of the dance they choreographed for them to perform. Most students also had to explore **control structures**, learning how to use KIBO’s Repeat Loop commands in order to ensure their robot dances repeated the appropriate number of times to match the music.

4.2. *Robotic Animals*

Integrating the natural sciences with robotics and engineering, in the *Robotic Animals* curriculum (DevTech, 2015), children explore animals and their natural habitats. After choosing an animal and researching its behavioral and physical characteristics, students create a robotic representation of that animal and its habitat for their final projects (See Figure 4).

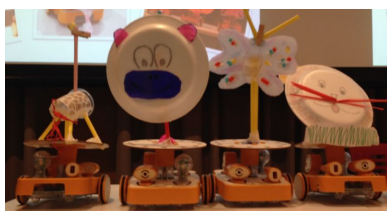


Figure 4. The image (left) shows final project examples from the *Robotic Animals* curriculum.

When building and programming their robotic animals, children grappled with the concept of **hardware and software**. They learned that to create an effective robot that looks, moves, and reacts like a cat or wolf, they needed to understand and use KIBO's hardware elements such as motors, sensors, and wheels as well as the right software, or program, to make the robot move the way the animal does. Children moved through an iterative **design process**, building their physical robot structure and made improvements to its sturdiness and aesthetic features. They also moved through an iterative process developing their programs.

4.3. *Patterns All Around*

The *Patterns All Around* unit (DevTech, 2015) focuses on an explicit exploration of math through KIBO. This unit integrates mathematics with fundamental engineering and programming concepts. Throughout the curriculum, students learn about different types of patterns using mathematics. They also explore other foundational math skills such as counting, shape recognition, and more. As a final project, students then have the opportunity to create a class “quilt” using large pieces of posterboard. By attaching a pen or crayon to KIBO, they were able to complete hands-on programming challenges where they were prompted to program KIBO to draw specific shapes or create different types of patterns on paper. This unit also offered many opportunities for free play and artistic exploration.

In this unit, students explored the computational concept of **modularity**, or breaking down a large task into a series of smaller steps. While programming complex patterns was often a daunting task for the kids, their teachers prompted them to focus on programming just one part at a time. After coming up with a series of short programs, children were able to put it all together and **debug**, or troubleshoot if it still did not look quite right.

5. RESEARCH WITH KIBO

5.1. *Methods*

During the research and development of KIBO, we have collected quantitative and qualitative data and published findings from $N=322$ children and $N=32$ early childhood teachers over the course of dozens of studies looking at what children have learned about robotics, engineering, sequencing, and more using KIBO (See Table 2 on the following page). Our research has been conducted across the United States, in Denmark, and as part of a large-scale study in Singapore (Sullivan & Bers, 2017). In order to measure children's mastery of computational concepts, the

DevTech Research Group developed the “Solve-Its” assessment (Strawhacker, Sullivan, & Bers, 2013; Strawhacker & Bers, 2014). Solve-Its entail listening to different stories or songs being read or sang aloud by a researcher. After listening to the story or song, the Solve-Its prompt children to arrange paper blocks into a sequential program that matches what they heard (See Figure 6). Each task assesses a different computational concept such as control flow or sequencing.



Figure 6. The image (above) shows an example of a child completed Solve-It task assessing their knowledge of repeats.

Highlights from this work are summarized in the following section. For a full list of publications detailing our studies with KIBO and to find out about the materials we have developed including teacher surveys, interview protocols, observation protocols, behavioral checklists, and more please visit: <http://ase.tufts.edu/devtech/publications.html>

Table 2. Summary of Topics Researched with KIBO

	Sample	Study Instruments
Sequencing	$N=27$	Baron-Cohen et al. picture sequencing cards
Computational Thinking	$N=28$	Solve-Its Debugging Assessment
Robotics & Programming Knowledge	$N=60$	Solve-Its Robot Parts Task
Gender	$N=45$	Solve-Its Interviews
Coding in Preschool	$N=64$	Solve-Its Observations
Teachers	$N=32$	Teacher surveys Interviews
Positive Technological Behaviors (PTD)	$N=98$	PTD Checklists
Total	$N=354$	

5.2. *What Do Children Learn?*

Our research has shown that learning to program with tangible robotics kits allows young children to practice sequencing, logical reasoning, and problem solving skills, along with positive behaviors such as collaboration and communication (Kazakoff, Sullivan, & Bers, 2013; Bers, 2015; Sullivan & Bers, 2015). In addition, we have shown that children as young as 4 years old can master powerful ideas from computational thinking and early engineering (Bers, 2017).

In a study with children in pre-kindergarten through second grade ($N = 60$) using a prototype of the KIBO robotics kit, results showed that beginning in pre-kindergarten, children were already able to master basic robotics and programming skills (Sullivan & Bers, 2016). This same study also demonstrated that older children were able to master increasingly complex concepts using the same kit in the same amount of time (Sullivan & Bers, 2016). Based on these findings, DevTech's most recent study with KIBO focused explicitly on the pre-kindergarten years and what these very young children are capable of building and creating (Elkin, Sullivan, & Bers, 2016). In this study, with 64 children from seven preschool classrooms, findings indicated that although KIBO was originally designed for ages 4 and up, children as young as age 3 could create syntactically correct programs for KIBO (Elkin, Sullivan, & Bers, 2016).

These findings demonstrated that KIBO embodies the "high ceiling/low floor" approach to technology design. This means it is easy to get started with KIBO (i.e. "low floor"), in this case, even for children as young as 3 years old. But there is also a "high ceiling" (i.e. a lot of complex possibilities) for what you can do with KIBO as you get older and gain more mastery for the concepts. Resnick, et al. (2005) also describes the idea of "wide walls" saying that "tools should support and suggest a wide range of explorations." In order to address this, the DevTech Research Group has created over a dozen curriculum units, such as the three described in the previous section, that explore the ways that robotics can be integrated across a variety of domains. Our research has demonstrated that it is not just children who need support and materials: teachers do too. We have seen that early childhood teachers need training, support, and resources in order to feel confident and competent teaching robotics (Bers, Seddighin, & Sullivan, 2013). Therefore, we have now made training videos, curriculum units, and other resources freely available on the Early Childhood Robotics Network (www.tkroboticsnetwork.ning.com).

5.3. Computational Thinking

A big piece of our research on computational thinking has focused on the impact of robotics and computer programming on young children's sequencing skills. Sequencing, a key aspect of computational thinking outlined by Bers (2017), is also an important pre-math and pre-literacy skill for early childhood found in both curricular frameworks and learning assessments (Kazakoff, Sullivan, & Bers, 2013).

Our research has demonstrated that beginning in pre-kindergarten, learning to program a robot significantly improves children's ability to logically sequence picture stories (Kazakoff, Sullivan, & Bers, 2013). This suggests that the sequencing skills gained through programming can be translated to sequencing things beyond code, such as stories.

In a recent study by Pugnali, Sullivan, & Bers (under review) the authors have begun to explore the impact of user interface on children's computational thinking skills, comparing the tangible KIBO programming language to a graphical tablet-based programming language. This study found that children in the tangible KIBO group scored significantly higher on two key aspects of computational thinking: sequencing and debugging. While further research is required, this may suggest that the tangible nature of KIBO's block language may make it more accessible to young children than onscreen languages.

6. CONCLUSION

The KIBO kit is being used by a growing number of children, parents, teachers, schools, camps, museums, and after school programs all around the world. Since its launch in 2014, KIBO is now used in 48 states across the U.S. as well as 43 countries worldwide. Countries such as Singapore are now using KIBO on a widespread basis to address technological literacy in the early childhood years (Sullivan & Bers, 2017). The research summarized here demonstrates the power of a tool like KIBO to effectively teach computational thinking beginning as early as pre-school and kindergarten. It also highlights the many ways that robotics and computer programming can easily integrate into traditional early childhood domains such as math, science, and social studies. Moreover, the work done with KIBO over the past five years has shown the possibilities for teaching computational thinking without forgetting that young children are still young children. Learning to code should not come at the sacrifice of learning to play and socialize. The curriculum units developed for KIBO have demonstrated successful ways to teach coding while still engaging in physical movement, listening to music, dancing, and collaborating. All of these are key components of a well-rounded early childhood experience.

7. REFERENCES

- American Academy of Pediatrics (2016). Media and young minds. *Pediatrics*, 138(5).
- Bers, M.U. (2017). *Coding as a Playground: Programming and Computational Thinking in the Early Childhood Classroom*. Routledge press.
- Bers, M. (2008). *Blocks to Robots: Learning with Technology in the Early Childhood Classroom*. Teachers College Press, NY, NY
- 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.
- Brennan, K., & Resnick, M. (2012). New Frameworks for Studying and Assessing the Development of Computational Thinking. Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada.

- DevTech (2015). *Dances from Around the World Curriculum*. Retrieved from: tkroboticsnetwork.ning.com
- DevTech (2015). *Robotic Animals Curriculum*. Retrieved from: tkroboticsnetwork.ning.com
- DevTech (2015). *Patterns All Around Curriculum*. Retrieved from: tkroboticsnetwork.ning.com
- Elkin, M., Sullivan, A., & Bers, M.U. (2016). Programming with the KIBO Robotics Kit in Preschool Classrooms. *Computers in the Schools*, 33(3), 169-186
- 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.
- Froebel, F. (1826). *On the Education of Man (Die Menschenerziehung)*, Keilhau/Leipzig: Wienbrach.
- Horn, M. S., & Jacob, R. J. (2007). Tangible programming in the classroom with tern. In *CHI'07 extended abstracts on Human factors in computing systems* (pp. 1965-1970). ACM.
- Kazakoff, E., & Bers, M. (2012). Programming in a robotics context in the kindergarten classroom: The impact on sequencing skills. *Journal of Educational Multimedia and Hypermedia*, 21(4), 371-391.
- Kazakoff, E., Sullivan, A., & Bers, M.U. (2013). The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. *Early Childhood Education Journal*, 41(4), 245-255.
- 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.
- Manches, A., & Price, S. (2011). Designing learning representations around physical manipulation: hands and objects. In *Proceedings of the 10th International Conference on Interaction Design and Children* (pp. 81-89). ACM.
- McNerney, T. S. (2004). From turtles to tangible programming bricks: Explorations in physical language design. *Personal and Ubiquitous Computing*, 8(5), 326-337.
- Montessori, M., & Gutek, G. L. (2004). *The Montessori method: The origins of an educational innovation: including an abridged and annotated edition of Maria Montessori's the Montessori method*. Lanham, MD: Rowman & Littlefield Publishers.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books.
- Piaget, J. (1936). *Origins of intelligence in the child*. London: Routledge & Kegan Paul.
- Perlman, R. (1976). *Using computer technology to provide a creative learning environment for preschool children*.
- Pugnali, A., Sullivan, A., & Bers, M.U. (under review). The impact of user interface on young children's computational thinking. *Journal of Information Technology Education: Innovations in Practice*.
- Resnick, M., Myers, B., Nakakoji, K., Shneiderman, B., Pausch, R., Selker, T., & Eisenberg, M. (2005). Design principles for tools to support creative thinking. *The School of Computer Science at Research Showcase at Carnegie Mellon University*.
- Russ, S.W. (2004). *Play in child development and psychotherapy*. Mahwah, NJ: Earlbaum.
- Singer, D.G. & Singer, J.L. (2005). *Imagination and play in the electronic age*. Cambridge, MA: Harvard University Press.
- Smith, A. C. (2007). Using magnets in physical blocks that behave as programming objects. In *Proceedings of the 1st international conference on Tangible and embedded interaction* (pp. 147-150). ACM.
- 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*. Online First.
- Sullivan, A. & Bers, M.U. (2017). Dancing Robots: Integrating Art, Music, and Robotics in Singapore's Early Childhood Centers. *International Journal of Technology & Design Education*.
- Sullivan, A., Elkin, M., & Bers, M. U. (2015). KIBO Robot Demo: Engaging young children in programming and engineering. In *Proceedings of the 14th International Conference on Interaction Design and Children (IDC '15)*. ACM, Boston, MA, USA.
- Suzuki, H., & Kato, H. (1995). Interaction-level support for collaborative learning: AlgoBlock—an open programming language. In *The first international conference on Computer support for collaborative learning* (pp. 349-355). L. Erlbaum Associates Inc..
- Svensson, A. K. (2000). Computers in school: Socially isolating or a tool to promote collaboration? *Journal of Educational Computing Research*, 22(4), 437-453.
- Wing, J. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-36
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical transactions of the royal society of London A: mathematical, physical and engineering sciences*, 366(1881), 3717-3725.
- Yadav, A. (2011). Computational Thinking and 21st Century Problem Solving.