Coding, Playgrounds and Literacy in Early Childhood Education: the Development of KIBO Robotics and ScratchJr

Marina Umaschi Bers
Tufts University
Medford, MA, USA
Marina.Bers@tufts.edu

***Abstract*—This paper describes two programming environments explicitly designed for early childhood education, the screen-based ScratchJr; and the tangible robotic kit KIBO. Both of these tools were explicitly designed to support the learning of concepts and skills of computer science and engineering in a developmentally appropriate way. The design principles are based on the notion of "coding as playground" and "coding as literacy".**

Keywords—coding; early childhood; robotics; computational thinking; programming; young children; education

I. 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. Children want to learn about the natural world and the artificial world, the world of emotions and the world of ideas, the world by themselves and the world with others in social contexts. In order for young children to master new knowledge, they need hands-on experiences to construct their learning [1].
Thus, early childhood classrooms integrate the range of learning experiences throughout the curriculum by exposing children to hands-on projects that integrate math, science, language, social studies, arts, and so forth.

With the advancement of novel interfaces, programming languages and robotics systems that are developmentally appropriate for young children have been developed over the last decade. These new technologies offer young children the possibility to also develop and integrate knowledge about computer science and engineering. In this paper I will describe the development of two of such introductory programming environments, ScratchJr, that runs on tablets and is programmed through a screen, and KIBO robotics kit that is programmed with wooden blocks (no screen or keyboards).

Both these programming environments were designed using two guiding principles to conceptualize children's experiences: 1) Coding as a playground and 2) Coding as literacy. Next, the paper will present these conceptual frameworks, followed by a description of ScratchJr and KIBO and its design features supporting the framework, as well as an overview of usage.

II. CODING AS PLAYGROUND

This paper presents an approach to coding as a playground. What is a playground? Playgrounds are environments designed to engage children in all domains of development (personal, social, moral, language, cognitive, motor, etc.) while having fun. Playgrounds are open-ended. They invite fantasy play, imagination and creativity, playing alone and with others, mastering skills and solving social conflicts. In contrast to the open-ended playground, playpens convey lack of freedom to experiment, lack of autonomy for exploration, lack of creative opportunities, and lack of taking risks. Although playpens are safer, playgrounds provide infinite possibilities for growth and learning [2]. Playgrounds are environments for discovery and learning.

The framework of **coding as playground** provides a way to understand the kind of developmentally appropriate experiences that programming environments must promote: problem solving, imagination, cognitive challenges, social interactions, motor skills development, emotional exploration, and making different personal and moral choices. Within this coding playgrounds, the programming language or symbol system to be manipulated, is a major component of the experience, but not the only one. Like in the playground, there are slides and swings and sand boxes and so forth.

Children have many options for things to do within the environments. Similarly, in ScratchJr, in addition to the icon-based programming blocks, there is a paint editor and sound-recoding system, and a library of characters and backgrounds to choose from. In KIBO, there are motors and sensors and art platforms as well as wooden blocks to sequence commands for the robot to follow once it is programmed.

978-1-5386-2957-4/18/$31.00 ©2018 IEEE 17-20 April, 2018, Santa Cruz de Tenerife, Canary Islands, Spain
2018 IEEE Global Engineering Education Conference (EDUCON)

Within these rich environments the experience of coding can become playful and creative. It offers many opportunities for learning and personal growth, exploration and mastery of new skills and ways of thinking. We do not always take children to the playground. There are other places to visit and other skills to develop. But when we do go to the playground, we want it to be a developmentally appropriate space. Same with programming environments for children. There are other kinds of technological environments to explore, games and apps. But when exposing young children to coding, playgrounds provide a powerful metaphor for the best kind of learning experiences.

The Positive Technological Development framework organizes these types of learning experiences around six behaviors or developmental milestones (6 C's) that need to be promoted while designing coding environments as playgrounds: Content creation, creativity, choices of conduct, communication, collaboration, and community building [2]. Coding can be a playground for children to become producers, and not merely consumers, in our technologically-rich world.

In the coding playground, young children create their own projects to communicate ideas and express who they are. They explore powerful ideas from computer science and engineering, they engage in problem solving and storytelling; they develop sequencing skills and algorithmic thinking. They journey through the design process from an early idea to a final product that can be shared with others. They also learn how to manage frustration and find a solution, rather than giving up when things get challenging. They develop strategies for debugging their projects. They learn to collaborate with others and they grow proud of their hard work. In the coding playground, children have fun while learning new things. They can be themselves and playfully explore new concepts and ideas, as well as develop new skills. They can fail and start all over again.

In this coding playground, children encounter powerful ideas from computer science that are useful not only for future programmers and engineers, but for everyone. Thus, the playground approach to coding moves the conversation beyond the traditional view of coding as a technical skill. Coding is a way to achieve literacy in the 21st century, like reading and writing.

III. Coding as literacy

Coding is a new literacy for the 21st century [3]. As a literacy, coding enables new ways of thinking and new ways of communicating and expressing ideas, as well as new ways of civic participation. Literacy ensures participation in decision-making processes and civic institutions. Those who can't read and write are left out of power structures. Their civic voices are not heard. Will this be the case for those who can't code? For those who can't think in computational ways?

I define literacy as the ability to use a symbol system (a programming language or a natural written language) and a technological tool (paper and pencil, or a tablet and computer) to comprehend, generate, communicate, and express ideas or thoughts by making a sharable product (a text, an animation, a robot) that others can interpret. This definition applies to both textual and coding literacy.

As a literacy, coding invites new ways of thinking (i.e. computational thinking) and carries the ability to produce an artifact detached from its creator, with its own meaning. There is a producer with an intention, with a passion, with a desire to communicate something. Coding, like writing, is a medium of human expression. Through this expressive process, we learn to think, feel and communicate in new ways.

It is in early childhood that the process of teaching to read and write begins [4,5]. Research shows that both from an economic and a developmental standpoint, educational interventions that begin in early childhood are associated with lower costs and more durable effects than interventions that start later on document the significance of early experiences for later school achievement [6, 7, 8, 9].

Only recently has there been emphasis on integrating computing in the early years due to the development of new programming interfaces, such as ScratchJr and KIBO, and a growing body of research that suggests children who are exposed to STEM curriculum and computer programming at an early age demonstrate fewer gender-based stereotypes regarding STEM careers [10, 11, 12, 13].

However, if coding is conceived as a literacy that must begin to be taught early in life, there is a need of programming environments explicitly designed to support the **coding as literacy** approach and that can be as developmentally appropriate as playgrounds. Later on, the paper will present both ScratchJr and KIBO as examples.

When exposed to programming environments children learn to master an artificial symbolic system (i.e. the programming language) to create interactive projects that can be shared with others [3]. Just like any natural language, English, Spanish, or Japanese, which allows us to express our needs and desires, our discoveries and frustrations, our dreams and everyday doings, programming languages provide a tool for expression. We need to learn their syntax and grammars and, over time, the more we engage with them, the more fluent we become. We know when we have truly learned a new language because we are able to use it for different purposes and to engage in computational thinking.

The term “computational thinking” can be defined as solving problems algorithmically and developing a sense of technological fluency [14, 15]. Children as young as four years old can learn foundational computational thinking concepts and this kind of learning can support their literacy, mathematical, and socio-emotional development [16, 17, 18, 19].While computational thinking is rooted in computer science, many have argued that it is a universally applicable attitude and skillset that fundamental for everyone to master, just like reading, writing, and arithmetic [14]. Coding engages and reinforces computational thinking. At the same time, computational thinking engages and reinforces coding.

Both KIBO and ScratchJr support the development of computational thinking. However, although the emphasis of both of these languages is put on sequential thinking, their design features support problem solving and expression by reducing unnecessary low-level cognitive burdens, freeing up mental resources for high-level processes such as troubleshooting a script that produces unexpected outcomes.

These design decisions keep the challenge at an appropriate level and may help young children devote sufficient cognitive resources to the many high-level thinking processes involved in imagining and creating a program. When the goal of learning to code is expression (as in literacy), and not only problem solving, this is much needed.

IV. ScratchJr

ScratchJr is a digital playground for coding. It provides a free introductory programming environment for young children ages 5-7. It was developed as a collaboration between the DevTech Research Group at Tufts University, the MIT Lifelong Kindergarten Group, and the Playful Invention Company, with funding from the National Science Foundation and the Scratch Foundation.

ScratchJr was first launched as a freely downloadable app on iPads in July, 2014, and has since been released for use on several other platforms including Android tablets, Amazon tablets, and Chromebooks. Used in classrooms and homes worldwide, ScratchJr enables children to create interactive stories and games by snapping together graphical programming blocks to make characters move, jump, dance, and sing. As shown in Figure 1, the ScratchJr interface allows children to use blocks that control motion, looks, sound, character communication, and more. Through these programming blocks, young children learn the basic concepts and powerful ideas of coding while creating personally meaningful projects [3].

Figure 1 ScratchJr interface

In December 2015 we launched PBS KIDS ScratchJr, in collaboration with PBS KIDS, so children could create their own interactive stories and games using over 150 characters and backgrounds from popular children's television programs produced by PBS KIDS. At the time of writing this paper, there is over one million downloads of this version.

ScratchJr has a user’s library of projects, a main project editor, and tools for selecting and drawing characters and background graphics. The blue palette of programming instructions lies along the center of the editor. Children display one instruction category at a time by clicking selectors on the left. Dragging instruction blocks from the palette into the scripting area below activates them. Snapping blocks together creates programs that are read and played from left to right. The “Green Flag” (“Play”) and red “Stop” buttons respectively start and interrupt the programmed animation.

The programming blocks are organized into six categories represented by different colours: yellow Trigger blocks, blue Motion blocks, purple Looks blocks, green Sound blocks, orange Control flow blocks, and red End blocks. When put together as a jigsaw puzzle, these programming blocks allow children to control their character’s actions on the screen.
The programming blocks span concepts from simple sequencing of motion to control structures

We started the design and development process of ScratchJr by observing how young children used Scratch, designed for older children 8 and up, and noting their difficulties. We spent many hours in local kindergarten, first, and second grade classrooms to understand the limitations for our intended age range, 5 to 7 years old [20]. For example, we noted that children were getting lost with so many possibilities for programing commands. Thus, we learned early on about the need to simplify and offer a more limited programming palette. We also noticed that movement happened too fast and children had a difficult time understanding the relationship between the programming blocks and their resulting actions. Thus, we decided to slow

down processes, so every block would take time before the triggering of the action.

We worked with hundreds of teachers and children through informal afterschool sessions, educator workshops, experimental classroom interventions, and at-home play sessions. Additionally, we conducted online surveys and face-to-face focus groups to obtain feedback. These provided valuable insights for our design team.

By July 2014, after a successful Kickstarter campaign to raise funds to complement our already existing NSF grant, we launched the first version of ScratchJr as a native tablet app.

A. ScratchJr's Design as a Playground

At the playground, children have diverse activities to choose from. They can go to the sand box, the swing, the slide or just run around. They can play with sticks, ride their bikes or create fantasy worlds. Similarly, while using ScratchJr, children can engage in all kinds of activities beyond coding. They can create and modify characters in the paint editor, record their own voices and sounds, and even insert photos of themselves that they take in the paint editor using the camera option. And, of course, they can incorporate those media rich materials into their projects to personalize them.

We worked in partnership with graphic designers so the interface would convey playfulness. The colour scheme sets a playful tone, the graphics are bright and whimsical, and the programmable actions are fun. Children drawn to artistic endeavours can design characters and backgrounds. Children drawn to animations can explore the range of programming concepts systematically or by tinkering.

However, ScratchJr comes with a small basic set of graphics compared to the hundreds available in Scratch. Just like playgrounds for younger children, offer limited play structures when compared to those for older children. This design decision was motivated by our overarching theme that "less is more" to ease children's difficulty in navigating vast arrays of options. Furthermore, it encourages children to create their own new graphics. Children can edit the included images or draw their own in an embedded scalable vector graphics editor. Data shows that 11% of all projects created by children include either a character or background that was created in the Paint Editor. Furthermore, the most common character added in ScratchJr, encompassing 30% of the characters added during sessions, is a character created or altered in some way in the paint editor. This demonstrates users' desire to add personal and unique aspects to their projects.

In terms of programming blocks, we also applied the "less is more" philosophy and carefully thought about including categories that would give flexibility, but not overwhelm children. The choice of bright colours and jigsaw puzzle shapes are explicitly are made to trigger the feeling of play and to engage children in playful behaviours when programming.

B. ScratchJr's Design as Literacy

The target age for ScratchJr is 5 to 6 years old, a time when children are learning how to read and write. In the US, and in many Western countries, writing happens in a sequence from left to right. Therefore we decided to mirror this directionality.

A programming script runs as a sequence from left to right instead of the traditional top-to-bottom format of most programming languages, including Scratch. This choice reinforces print-awareness and English literacy skills. As a character's script runs, the app highlights each block as it is executed, representing the instructions given to that character on the stage. Text showing the name of each block can be revealed by tapping it, which supports word recognition.

The design of the ScratchJr's block shapes prevents syntax errors. The jigsaw puzzle pieces have visual properties that correspond to their syntactic properties. For example, the "Repeat Forever" block can only appear at the end of a program. Since nothing should follow a "Repeat Forever" command, the right side of this block is rounded so that another jigsaw piece cannot attach to it.

Furthermore, several design decisions were made for seamless integration with literacy. The ability to create up to four independent "pages" and to integrate text and speech into a project allows children to create their own storybooks with a beginning, middle, and end. When creating these projects, children think in terms of "if this happens, then this happens." By programming with ScratchJr, they can begin to understand the basic components of a story while also reinforcing sequencing skills.

C. ScratchJr's usage over the world

Since its launch, ScratchJr has been downloaded 9.5 million times and has actively ben used in every country in the world (except North Korea and Western Sahara). The most usage happens in the following countries: United States (37%), United Kingdom (15%), Canada (7%), Australia (6%), Sweden (5%), France (4%), Spain (3%), Finland (2%), Netherlands (2%), and China (2%).

The ScratchJr team began collecting analytics data in January 2016, which has provided a more comprehensive overview of how children and educators are using the application. Since then, as of February 2018, over 19 million projects have been created, and over 26 million existing projects have been opened again and edited, indicating that users are working on improving and altering the same projects over time. Additionally, over 600,000 projects have been shared with others via email or Apple

AirDrop. In this relatively short time span, over 406 million programming blocks have been used, the most popular blocks being "Forward," "Start on Green Flag," "Up," "Backward," and "Say." The "Say" block is used so that characters within a project can communicate with each other—the notion that it is in the top 5 most used blocks on ScratchJr shows that children are using the app to build upon their storytelling skills. Additionally, ScratchJr maintains a rate of 249,000 returning users each month, while still bringing in a consistent rate of 255,000 new users each month.

Throughout the world teachers use ScratchJr to teach literacy, storytelling, and creative expression. Additionally, ScratchJr is integrated into many different curricular units such as drama, math, and social studies.

While ScratchJr provides a screen-based playground, KIBO robotics, described next, provides a tangible experience.

V. KIBO ROBOTICS

KIBO is a robot kit specifically designed for children ages 4-7 years old. Young children learn by doing, thus KIBO provides opportunities for doing different things. Children can build their own robot, program it to do what they want, and decorate it with art supplies. KIBO gives children the chance to make their ideas physical and tangible—without requiring screen time from PCs, tablets, or smartphones.

The concept, prototypes, and research for KIBO were born in the DevTech research group in 2011 through generous funding from the National Science Foundation. KIBO's design was based on years of research in collaboration with teachers and early childhood experts to meet the learning needs of young children in a developmentally appropriate way [21, 22, 23]. Later on, KIBO became commercially available worldwide in 2014 through KinderLab Robotics (see www.kinderlabrobotics.com).

As a robotics construction kit, KIBO has hardware: the robot body, wheels, motors, a light output, a variety of sensors, and art platforms, and software composed of a tangible programming language made of interlocking wooden blocks (see figure 2). 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. These wooden blocks contain no electronic or digital components. Instead, the KIBO robot has an embedded scanner. The scanner allows users to scan the barcodes on the wooden blocks and send a program to their robot instantaneously.

Figure 2: The KIBO robotic kit

KIBO's design builds on extensive research on tangible programming that use physical objects to represent the various aspects of computer programming [24, 25, 26, 27, 28, 29, 30].

In addition to the tangible programming language, the KIBO robot comes with sensors and actuators (motors and light bulb), as well as art platforms. These modules can be interchangeably combined on the robot body. The use of sensors is well aligned with most early childhood curriculum that engages children in exploring both human and animal sensors. Three motors are also included with the robot, two can be connected to the opposite sides, for mobility, and one motor can be located on top, for rotation of an attached element such as the art platform.

KIBO went through many prototypes before it made it out into the world. At every stage of development and testing we collaborated with early educators, children, and specialists to create an age appropriate robotic kit that would be intuitive and engaging, but also challenging [17, 21].

The first KIBO prototype (called "KIWI") was programmed with CHERP and required the use of a computer and a webcam to take pictures of the blocks and send the program to the robot through a USB cable [30].

We hand built 10 first prototypes and tested them in focus groups, professional development workshops, and the classroom. Data was collected from 32 early childhood educators in 2013 on their attitudes, opinions, and experiences to inform the re-design of the prototype [33].

Teachers were drawn to the use of wood and the simplicity of our first design. However, they indicated that programming the robot should use minimal or no computer equipment. Although KIBO was successful in engaging children in foundational programming skills, the robotic parts were too easy to assemble and did not sufficiently engage children in engineering problem-solving or creative artistic design.

Informed by this, we built our second prototype with a 3D printed KIBO body including an embedded scanner, eliminating the need to use a computer. This directly addressed the teachers' concerns about availability of

computers in early childhood classrooms and screen-time for young children. Additionally, instead of having a "magical black box" with the electronic components inside the robot's body, the newer KIBO prototype had a clear plastic bottom, allowing children to see the wires, batteries, microprocessor, and other parts involved in making the robot function.

Furthermore, in order to address the need of more "engineering," we designed the wheels to connect to the motors in two different ways, prompting children to test how changing the wheels' orientation makes the robot move differently. The shapes and looks of both sensors and light output components were also redesigned to address earlier pilot testing with children. Each sensor's aesthetic is designed to convey meaning (i.e. the ear-shaped part is a sound sensor; the eye-shaped part is a light sensor; and the telescope-shaped part is a distance sensor). During the late pre-operational stage of cognitive development (ages 4-6), children extend and apply culturally-learned symbol systems to interactions with the physical and social world; thus, the explicit emphasis on design features with symbolic representations.

The aesthetic features of KIBO have remained purposefully plain throughout each prototype iteration. The "unfinished" look invites children to complete the robot using their own imaginative creations. Much like a blank canvas or un-sculpted clay, KIBO inspires children to add to it. This supports a variety of sensory and aesthetic experiences.

A. KIBO's Design as a Playground

KIBO was designed to be a playground for tangible coding in which children could encounter powerful ideas from computer science and develop computational thinking, while engaging in six positive behaviors identified in the playground: content creation, creativity, choice of conduct, communication, collaboration and community building [2].

KIBO supports children in making almost anything: a character from a story, a carousel, a dancer, a dog sled. The possibilities are endless, as wide as children's own imaginations. The child puts together a sequence of instructions (a program) using the wooden KIBO blocks. Then, they scan the blocks with the KIBO body to tell the robot what to do. Finally, they press a button and the robot comes "alive." KIBO engages children in becoming programmers, engineers, problem solvers, designers, artists, dancers, choreographers, and writers.

The choice of having no computer, tablet, or other form of "screen-time" to be required to program with KIBO, was an attempt to make the coding experience closer to a playground experience. This design choice is also aligned with the American Academy of Pediatrics' recommendation that young children should have a limited amount of screen time per day [34].

The use of wooden blocks for KIBO was inspired by the tradition of learning manipulatives already used in early childhood classrooms to teach shapes, size, and colors in a playful way [35, 36].

Finally, the physicality of KIBO invites children to use their bodies, like in the playground, and engage in motor activities, while also collaborating with others. The size of the robot allows it to be shareable to promote social interaction. This allows children to engage in problem solving in a developmentally appropriate way. Furthermore, it promotes a journey towards computational literacy that enables the use of the robot for personal expression and communication.

B. KIBO's Design as Literacy

KIBO was designed to support early literacy development and to be easily integrated with language learning classes. The programming language pairs iconic images and simple words. Actions are labeled with text, as a way to scaffold early letter and word recognition. Furthermore, individual commands (wooden blocks) are designed to be put in a sequence from left to write, just like letters into words, and words into sentences.

With KIBO, children arrange and connect wooden blocks to give commands to their robots. Every sequence starts with a green begin block, like a capital letter, and ends with a red end block, like a period. Only after the sequence, or sentence, is properly constructed, can KIBO complete scan the blocks.

The physical properties of the wooden blocks are exploited to express and enforce syntax. For example, the KIBO begin block doesn't have a hole, only a peg, because there is nothing that can be placed before the begin; and the end block doesn't have a peg, because there is no instruction that can go after the program ends. The language syntax in KIBO (i.e. a sequential connection of blocks) is designed to support and reinforce sequencing skills in young children, which are fundamental for later academic success in literacy.

In early childhood, one of the goals of literacy is to increase children's vocabulary. KIBO offers an opportunity to introduce new words from the technical domain and its use, guided by the playground approach, engages children in applying that new vocabulary words, communicating with teachers and peers, and writing and drawing notes in their design journals.

The ultimate goal of literacy is to build a student's comprehension and writing skills, so students can become creative and critical thinkers as well as avid communicators. This goes beyond mastering syntax and grammar. Similarly, the design of KIBO is aimed at shifting the problem-solving focus away from low-level problems (i.e., syntax and

connection errors) towards high-level problem solving so children can create robotic projects to communicate their ideas and express themselves.

C. KIBO in the world

Since its launch in 2014, KIBO has made it into private and public schools, museums and libraries, after school programs and summer camps, both in the US and abroad. As of today, KIBO can be found in 51 countries and is used by thousands of teachers in over 500 schools, as well as in thousands of homes, libraries, and enrichment programs. KIBO has been used to teach a variety of curricular topics ranging from science to literacy, geography to religion, as well as to engage children in developing social-emotional skills. Pilot studies have been done with children on the autism spectrum and with blind children [37]. For example, in Utah Schools for the Deaf and the Blind, teachers put Braille labels on the blocks and are successfully teaching with KIBO.

At a local public school in Somerville, Massachusetts, early childhood teachers implemented a robotics curriculum in grades K-2 with the goal of fostering prosocial behaviors and community building. Students also programmed their robots to demonstrate respectful behaviors and school rules that kids often forget. For example, one student created a robot that reminded students to listen to their teacher during circle time. Another student created a robot to demonstrate walking through the hallway quietly and making lots of noise only when it reached the playground.

In a summer camp, KIBO was used to bring literacy and the arts to life in a playful way. In this one-week camp, students read a different book each day and programmed their robots to "act out" their favorite scenes from each book as well as alternative endings. As a final project, children read the iconic children's book *Where the Wild Things Ar*e by Maurice Sendak [38]. Children designed and built their own KIBO monsters inspired by the story and programmed them to act out the "Wild Rumpus" scene (i.e., a wild monster party).

In Singapore, preschool classrooms used KIBO to explore the rich multilingual and multicultural heritage of the students. Over the course of 7-weeks, students learned about robotics and programming through a KIBO curriculum called *Dances from Around the World*. For their final projects, students built and programmed robots that performed dances from the cultural backgrounds present in Singapore. For example, one group of students created a Lion Dancing robot to represent Chinese heritage while another group programmed KIBO to dance along to a Malay song [39]. A small-scale version of this project was implemented in Tenerife, Spain, and results are being analyzed. In sum, research with KIBO shows that children as young as four are capable of successfully building and programming robots. As children get older, they are able to master increasingly complex concepts including repeat loops, sensors, and conditional statements. Children are able to use KIBO to explore a variety of interdisciplinary content including literature, history, and more. Finally, research with KIBO shows that young children are able to practice social-emotional skills such as communication and collaboration when using the KIBO robotics kit.

VI. CONCLUSION

As more people learn to code and computer programming leaves the exclusive domain of computer science and become central to other professions, the civic dimension of coding as literacy comes into play. Literacy has the power to bring about social change.

Literacy ensures participation in decision-making processes and civic institutions. From an historical perspective, and currently in the developing world, those who can't read and write are left out of power structures. Their civic voices are not heard. Will this be the case for those who can't code? For those who can't think in computational ways?

However, we don't expect every child to grow into a professional writer. Textual literacy is both an important skill and intellectual tool for everyone. So is with coding. Not all children need to grow into software engineers and programmers, but they need computational literacy so they can become producers, and not only consumers, of digital artifacts. However, for children to be able to code, they need developmentally appropriate programming languages, such as KIBO and ScratchJr that invite the learning of abstract, logical thinking in a playful way.

VII. ACKNOWLDEGMENTS

I am grateful to the wonderful team of students and staff at the DevTech Research group at Tufts University, the Scratch foundation and the National Science Foundation for financially supporting this work.

REFERENCES

[1] J. Piaget, Origins of intelligence in the child, London: Routledge & Kegan Paul, 1936, [La naissance de l'intelligence chez l'enfant].
[2] M.U. Bers, Designing digital experiences for positive youth development: from playpen to playground, Cary, NC: Oxford, 2012.
[3] M.U. Bers, "Coding as a literacy for the 21st century," Education Week, 2018.
[4] Learning First Alliance, "Every child reading: An action plan of the Learning First Alliance", 1998.
[5] K. McIntosh, R.H. Horner, R.J. Chard, J.B. Boland, and R.H. Good, "The use of reading and behavior screening

measures to predict non-response to school-wide positive behavior support: A longitudinal analysis," School Psychology Review, vol. 35, pp. 275-291, 2006.
[6] F. Cunha, and J. Heckman, "The technology of skill formation," American Economic Review, vol. 97, no. 2, 2007, pp. 31–47, 2007.
[7] J. Heckman and D.M. Masterov, "The productivity argument for investing in young children," Working Paper 5, Invest in kids working group, Center for Economic Development, 2004.
[8] National Research Council, "Eager to learn: Educating our preschoolers," Infant and Child Development, vol. 11, Washington, DC: National Academy Press, pp. 283-284, 2001.
[9] National Research Council, "From neurons to neighborhoods: The science of early childhood development," Washington, DC: National Academy Press, 2002.
[10] S. S. Metz, "Attracting the engineering of 2020 today," in R. Burke, M. Mattis, and E. Elgar (Eds.), Women and minorities in science, technology, engineering and mathematics: Upping the numbers, Northampton, MA: Edward Elgar Publishing, pp. 184–209, 2007.
[11] C. M. Steele, "A threat in the air: How stereotypes shape intellectual identity and performance," American Psychologist, vol. 52, 1997, pp. 613–629.
[12] H. Madill, R.G. Campbell, D. M. Cullen, M.A. Armour, A.A. Einsiedel, A. L. Ciccocioppo, et al., "Developing career commitment in STEM-related fields: Myth versus reality," in In R. Burke, M. Mattis, and E. Elgar (Eds.), Women and minorities in science, technology, engineering and mathematics: Upping the numbers, Northhampton, MA: Edward Elgar Publishing, 2007, pp. 210–244.
[13] L. R. Markert, "Gender related to success in science and technology," The Journal of Technology Studies, vol. 22, no. 2, 1996, pp. 21-29.
[14] J. M. Wing, "Computational thinking," CACM Viewpoint, vol. 49, no.3, pp. 33-35, 2006 March. [Online]. Available: http://www.cs.cmu.edu/afs/cs/usr/wing/www/publications/Wing06.pdf. [Accessed Feb. 7, 2018]
[15] S. Papert, Mindstorms: Children, computers, and powerful ideas, New York, NY: Basic Books, 1980.
[16] M.U. Bers, Coding as a playground: programming and computational thinking in the early childhood classroom, Routledge Press, 2017.
[17] M. Bers, Blocks to robots: Learning with technology in the early childhood classroom, New York: Teacher's College, 2008.
[18] E. Kazakoff and M. Bers, "Programming in a robotics context in the kindergarten classroom: The impact on sequencing skills," Journal of Educational Multimedia and Hypermedia, vol. 21, no. 4, 2012, pp. 371-391.
[19] E. Kazakoff, A. Sullivan, and M.U. Bers, "The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood," Early Childhood Education Journal, vol. 41, no. 4, 2013, pp. 245-255.
[20] L.P. Flannery, E.R. Kazakoff, P. Bontá, B. Silverman, M.U. Bers, and M. Resnick, "Designing ScratchJr: Support for early childhood leanring through computer programing," in Proceedings of the 12th International Conference on Interaction Design and Children, IDC '13, ACM, New York, NY, USA, pp. 1-10, 2013.
[21] A. Sullivan, M. Elkin, and M.U. Bers, "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, 2015.
[22] A. Sullivan and M.U. Bers, "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, 2015.
[23] E. R. Kazakoff and M.U. Bers, "Put your robot in, Put your robot out: Sequencing through programming robots in early childhood", Journal of Educational Computing Research, vol. 50, no. 4, 2014.
[24] A. Manches and S. Price, "Designing learning representations around physical manipulation: Hands and objects," in Proceedings of the 10th International Conference on Interaction Design and Children, ACM, pp. 81-89.
[25] R. Perlman, Using computer technology to provide a creative learning environment for preschool children, [Logo memo no. 24]. Cambridge, MA: MIT Artificial Intelligence Laboratory Publications 260, 1976.
[26] H. Suzuki and H. Kato, "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., 1995.
[27] T.S. McNerney, "From turtles to tangible programming bricks: Explorations in physical language design," Personal and Ubiquitous Computing, vol. 8, no. 5, 2004, pp. 326-337.
[28] P. Wyeth and H.C. Purchase, "Tangible programming elements for young children," in CHI '02 Extended Abstracts on Human Factors in Computing Systems, ACM 2002, pp. 774-774.
[29] M.S. Horn and R. J. Jacob, "Tangible programming in the classroom with tern," in CHI '07 Extended Abstracts on Human Factors in Computing Systems, ACM, 2007, pp. 1965-1970.
[30] M.S. Horn, R.J. Crouser, and M.U. Bers, "Tangible interaction and learning: the case for a hybrid approach," Personal and Ubiquitous Computing, vol. 16, no. 4, 2012, pp. 379-389.
[31] M.U. Bers, S. Seddighin, and A. Sullivan, "Ready for robotics: Bringing together the T and E of STEM in early

childhood teacher education," Journal of Technology and Teacher Education, vol. 21, no. 3, 2013,pp. 355-377.
[32] American Academy of Pediatrics, "American Academy of Pediatrics Announces New Recommendations for Children's Media Use," 2016. [Online]. Available: https://www.aap.org/en-us/about-the-aap/aap-press-room/pages/american-academy-of-pediatrics-announces-new-recommendations-for-childrens-media-use.aspx. [Accessed: Feb. 7, 2018].
[33] F. Froebel, On the Education of Man, Keilhau/Leipzig: Wienbrach, 1826 [Die Menschenerziehung].
[34] M. Montessori and G. L. Gutek, 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, 2004.
[35] J.A. Canals, A. Barco, E. Relkin, D. Hannon, M. Heerink, M. Heinemann, K. Leidl, and M.U. Bers, The use case of KIBO robot to positively impact social and emotional development in children with ASD, unpublished.
[36] M. Sendak, Where the wild things are, New York: HarperCollins Publishers, 2013.
[37] A. Sullivan and M.U. Bers, "Dancing robots; Integrating art, music, and robotics in Singapore's early childhood centers," International Journal of Technology and Design Education, 2017.