Big Robots for Little Kids:

Investigating the Role of Scale in Early Childhood Robotics Kits

A thesis submitted by

Miki Z. Vizner

In partial fulfillment of the requirements of

Master of Arts

in

Child Study and Human Development

TUFTS UNIVERSITY

August 2017

Committee Members

Dr. Marina Umaschi Bers (Chair)
Eliot-Pearson Department of Child Study & Human Development

Dr. George Scarlett
Eliot-Pearson Department of Child Study & Human Development

Dr. Brian Gravel
Department of Education

Abstract

Couch fort and refrigerator box constructions are staples of early childhood play in American culture. Can this this large-scale fantasy type of play be leveraged to facilitate computational thinking? This thesis looks at the ways Kindergarteners (age 5-6) use two variations of the KIBO robotics platform in their play and learning. The first is the standard KIBO kit developed at the DevTech research group at Tufts University and commercialized by Kinderlab robotics. The second, created by the author, is 100 times bigger and can be ridden by children and adults. Specifically this study addresses the research question "How are children's experiences with big-KIBO different from KIBO?" To do so this thesis presents two analytical tools that were assembled conceptually from literature and the authors experiences with KIBO, examined using the data collected in this study, refined, and used as frameworks for understanding the data. They are a developmental model of programming with KIBO and an operationalization of Bers's (2018) powerful ideas of computational thinking when using KIBO. Vignettes from the data are presented and analyzed using these frameworks. Content and structural play themes are extracted from additional vignettes with each robot. In this study there are no clear differences in the ways children engage in computational thinking or develop their ability to program. There appear to be differences in the ways children play with the robots. Suggesting that a larger robot offers new opportunities and pathways for children to engage in computational thinking tasks. This study makes a case for the importance of thinking developmentally about computational thinking. Connections to literature and theory as well as suggestions for future work, both for children and designers, are discussed.

Keywords: robotics, technology, early childhood education, computational thinking, play, maker space

Acknowledgments

I owe great thanks to the group of people who have mentored, supported, and "made" with me over the course of this project. I would like to begin by thanking my committee. Marina Umaschi Bers as the chair of my committee and my advisor at Eliot Pearson you have shown me how to be a maker and a developmentalist at the same time. KIBO and Scratch Jr's global impact is inspiring; it is my goal to one day create an object as empowering. Thank you for your honest feedback and for believing in me as I move through this process and towards that goal. George Scarlett you have shown me that it is possible to scope, order, and artfully explain the complex puzzle that is human experience. Thank you for your patience and for pushing me to focus on the organization of acts, I thoroughly enjoyed it. Brian Gravel your mentorship has been a great factor in my own development throughout my master's experience. Our time at Malden has shown me how academia can have profound impact on the lives of individuals and communities. Thank you for supporting me through this journey, it has been a privilege to work together for the past 2.5 years.

The DevTech research group has been the source of much academic stimulation, work, and laughter. Kaitlyn Leidl thank you for sharing "the stage" with me and welcoming me as a member of the lab. Amanda Sullivan you are a star academic, early childhood educator, undergraduate educator, and teacher educator. Your ability to synthesize and move across these worlds is impressive; I have greatly appreciated you sharing these perspectives with me. I could not have asked for a better research partner than Amanda Strawhacker. It has been a pleasure to think, write, and "make" with you, thank you. I am excited to see where the early childhood maker space project will go and what you will uncover in your PhD work!

While there are many makers that have helped me along the way there are three that deserve special mention. Jon Evans, thank you for your technical advice, what would a thesis at Tufts be with out you? Graham Yeager for the one-shot 13 hour welding session that produced big-KIBOs frame, thank you. Rob Hart, thank you for welcoming me to the How to Make (almost) Anything Community at Harvard and beyond. Now that this is over we can all get back to “making” together!

I would like to thank the CEEO Innovation Fund and the LEGO Foundation for funding this project, and the Evelyn G. Pitcher Curriculum Lab for housing and supporting the early childhood makerspace in which my study was conducted.

Finally I would like to thank all of the children who participated in this study. Without you I would just be a grad student with a big robot. Your convictions, curiosity, and imagination are inspiring. Don’t let them go. This work is for you.

TABLE OF CONTENTS

List of Tables

List of Figures

Chapter One: Introduction

After thirty minutes of hard work Jane has finally attached the pillowcase sail to the top of its curtain rod mast. A ship constructed out of couch cushions, bedding, and other odds and ends is moored at the center of the living room, ready to set sail. This type of activity is common-place in early childhood (K-2) where play is at the center of the work being done. Jane could certainly entertain herself for hours over several days on this stationary ship, but what would change if there was a way to make it sail around the living room?

There are many robotics platforms for young children, but none of them are designed to handle the scale of Jane's project. I have developed a large-scale robotics kit, based on the KIBO robotics platform to leverage this type of big construction already fused with fantasy play as an opportunity to engage young children in computational thinking.

KIBO was developed by the DevTech lab at Tufts University and is specifically designed for early childhood (ages 4-7). KIBO has been the focus of a great deal of scholarship (Bers, 2018; Sullivan, Bers, & Mihm, 2017; Sullivan & Bers, 2017; Elkin, Sullivan, & Bers, 2016; Sullivan, Elkin, & Bers, 2015) centering on both the importance of educational robots in early childhood, as well as, how to design for and teach programming and computational thinking to the youngest learners. A full description of KIBO is presented in chapter four.

The DevTech lab at Tufts University has placed an emphasis on designing digital playgrounds to promote exploration and problem solving through educational technologies. While Bers's (2012) work has focused on extending the metaphor of playgrounds to the digital world, I aim to bridge the gap between these digital playgrounds and those in the world by looking at large scale interactive objects. This thesis begins this task by exploring children's interactions with a version of the KIBO robot large enough for them to ride.

Chapter Two: Review of the literature

Constructionism

As a budding constructionist, it seems appropriate that I begin this section with an experience rather than a definition. Last year, as part of my course work for *Children and New Technologies (CD 114)* taught by professor Bers, I developed the Curious Construction Kit, (Vizner & Strawhacker, 2016) a collection of connectors and large cardboard panels with embedded electronics that children could shape with their imaginations and build with their whole bodies. One child, Adam, while playing with the sound panel and a slider to create variable tones realized that he was acting as a DJ. Excitedly he exclaimed, “I’ve never been a DJ before!” Another, Sara, diligently assembled a house. It was important that it had four walls, a roof, and a door. When it was complete she entered and said, “this is my house.”

In this example we see two children creating their own knowledge of DJness, and houseness. Constructionism proposes that technologies have powerful educational applications “when used for supporting the design, the construction, and the programing of personally and epistemologically meaningful projects” (Marina Umaschi Bers, 2008). Both Adam and Sara worked on their projects because they were personally meaningful to them. This allowed them to dive into the disciplinary aspects of their work. Sara might have asked herself “how can I get this house to stay up?” and spend time working out how different angles yielded different results. Adam was exploring cause and effect by moving the slider and listening to the different sounds.

There is something different in the ways that Adam and Sara went about their work. Sara began with a house, and all of its components in mind. Through mucking about Adam was transformed into a DJ. These two styles are discussed in the literature (Turkle & Papert, 1990; Papert & Harel, 1991;Worsley & Blikstein, 2013; Resnick, 2006; Vizner & Strawhacker, 2016) as a “hard” or top-down systematic approach, in the case of Sara, and a “soft” bricolage approach, as represented by Adam. Both are important and should be nurtured in any learning context (Resnick, 2006).

The notion of Constructionism stems from the more widely known Piagetian theory of Constructivism, which is concerned with the individual’s building of knowledge structures. While a full description of Constructivism is outside the scope of this paper what I would like to address are the notions of accommodation and assimilation. “Assimilation refers to the tendency to interpret experience through existing structures of knowing as much as possible and is assumed to be the child’s initial tendency, while accommodation refers to the realization that current structures are insufficient for adequate understanding and that they must be changed.” (Feldman, 2004). Accommodation is far more challenging, the child must abandon their working theory of the world. This happens more robustly when the child is engaged in building a public, shareable artifact (Blikstein, 2015; Papert, 1993).

Piaget’s Constructivism describes the processes by which knowledge is constructed in the mind through an individual’s cognitive processes given a social and cultural context. Papert’s Constructionism extends Piaget’s ideas by paying particular attention to constructions in the world as supports for those in our minds (Marina Umaschi Bers, 2008).

Both Constructivist and Constructionist ideas have permeated many educators’ and designers’ thinking. Both Mariah Montessori and Loris Malaguzzi—founder of

Reggio Amellia educational philosophy—were influenced by Piaget's constructivism (Elkind, 2003; Hewett, 2001) and have a rising impact on schools in North America (Edwards, 2002). Not only have Constructivism and Constructionism been powerful influences on pedagogies, but they have also played a role in the design of many education tools. Papert's LOGO programming language and the LEGO Mindstorms robotics kit— which were heavily influenced by his ideas—as well as a suite of technologies created in their wake, have created millions of child programmers and roboticists around the world ("In Memory," n.d.)—myself included.

Educational Robots

In fifth grade my friend Jason and I spent weeks of our extra time working on a lunar lander video game in the computer lab. The longer we worked on it the more complex it got. At first you could fly the spaceship around, using the arrow keys, at will. Next we added gravity so that the space ship would always travel in a downward motion. After that we created different maps with landing pads and difficult terrain. If you crashed your ship would explode. At times for our amusement we could flip gravity and play the game backwards, or, super strong making the game nearly impossible. While this game wasn't novel to us, the making of it was. You might be asking yourself, "how could two fifth graders in the 90s possibly make their own video game?" We were using a LOGO programming environment called Microworlds.

In the late 1960s Papert introduced LOGO and for the first time children were put in control of computers. "The child programs the computer. And in teaching the computer how to think, children embark on an exploration about how they themselves think". LOGO allows children to program Turtles, "computer-controlled cybernetic animals," for programming and thinking with. Originally there were two versions of this turtle, one that lived in the computer that could be programmed to make shapes on the screen, and the other, a robot, a physical object that could be picked up like any other toy. (Papert, 1993)

These turtles create fertile ground for children to explore abstract concepts in concrete ways. Jason and I were programming turtles; they enabled us to explore ideas about gravity and acceleration, whose mathematical representations would otherwise be inaccessible to fifth graders (Papert, 1993). We were able to play with programming and more importantly fix "bugs." "The question to ask about the program is not whether it is right or wrong, but if it is fixable" (Papert, 1993). This mindset about learning and thinking can be applied to other domains of knowledge, transforming the fear of "being wrong" into a bug to be fixed.

In the 1980s a team at MIT began developing LEGO/Logo that allowed children to build machines out of traditional LEGO bricks as well as newly designed gears, motors, and sensors that could be programmed with LOGO. "LEGO/Logo represented a return to Logo's roots" as a physical object to be programmed (Blikstein, 2015). These new physical devices made the processes of making abstract concepts concrete even more powerful (Marina Umaschi Bers, 2008).

These new physical devices extended the ways in which children could engage with personally meaningful projects. They provide opportunities for both the "hard" and "soft" designers (Turkle & Papert, 1990). Some children may get lost in the power that playing with gears provides. While others may use these tools to construct a cast of robotic creatures and invest heavily in telling a story. In either case these children connect to the powerful ideas of programming through personally meaningful projects. Becoming technologically fluent as they express themselves with ease like one does with language (Marina Umaschi Bers, 2008).

Tangible programing languages were first introduced by Radia Perlman, a researcher in the MIT Logo lab in the 1970s. She believed that syntax rules of computer based programming led to barriers for children to be able to engage with the powerful ideas they held (Perlman, 1976). Over the years many tangible user interfaces (TUI) have been developed for a variety of applications. TUIs have been used to control LEGO WeDo, KIWI and KIBO robotics kits, and have shown to be effective for children as young as four to engage with programming (Strawhacker, Sullivan, & Bers, 2013; Bers, Seddighin, & Sullivan, 2013). Of these kits the KIBO robot is unique in that it uses wooden blocks as a TUI, drawing on a paradigm children are already familiar with for making meaning. The use of manipulative in early childhood robotics allows children to develop their sensori-motor skills, which are equally important, along side their cognitive skills (Bers, 2008). TUIs also provide children with the ability to engage with programming with out the use of screens (Bers & Horn, 2010), which is a growing concern in our society ("American Academy of Pediatrics Announces New Recommendations for Children's Media Use," n.d.).

So far this discussion has been about how children think *with* robots, I'd like to extend it by addressing what children think *about* robots, for even young children are deeply reflective in their thinking. The first chapter of Sherry Turkle's (1984) book *Second Self* discusses children's views on the aliveness of computer toys. Through interviewing hundreds of children from age 4-14 she shows how in the earlier years (younger than 8) children respond to the question of aliveness by offering behavioral rationales such as “it can talk, so it must be alive.” Older children tend to offer psychological rationales for why such toys are alive “computers can remember, they must be alive” or not “computers are smart, but they don’t have feelings.” More important than whether children believe robots are alive or not is the quality of the conversation these objects evoke (Turkle, 1984) and what they do with them. The ability to think *with* robots is inextricably intertwined with what one thinks *about* them.

Computational Thinking

Computational thinking refers to a broad set of analytical and problem solving skills, design principles, and practices derived from computer science (Wing, 2006). Doing computational thinking means that one expresses solutions, problems, and ideas in a form that a computer could execute.

Ken, my first computer science teacher had two mantras he drilled in to our fifth grade heads. The first was, “computer science is a paper and pencil exercise.” Computational thinking is something that humans do to leverage the computing power of machines (Wing, 2008).

In her new book *Coding as a Playground*, Marina Bers (2018) identifies seven powerful ideas of computational thinking for early childhood: algorithms, modularity, control structures, representation, hardware/software, design process, and debugging. Table 1 presents these powerful ideas. Bers reminds us that these ideas develop over the course of early childhood and beyond and may look different in different children, at different ages, and in different contexts.

Table 1: *Powerful Ideas of Computational Thinking from Bers (2018)*

Powerful Idea	Description
CONCEPTS	
Algorithms	"Series of ordered steps in a sequence to solve a problem or achieve some goal." Sequencing is an important skill in early childhood. It is at the core of being able to tell a story, tie one's shoes, and make a peanut butter and jelly sandwich.
Modularity	The decomposing of a complex task or procedure into more manageable sub-parts, and an understanding that sub-parts can be put together to make a more complex entity. The ability to use sub-parts from a solution to one problem with other sub-parts to solve a new problem.
Control structure	The initiation and order of execution of a set of commands. This includes repeats, loops, conditionals, events and nested structures. Making decisions based on a set of conditions. For example when a button is pressed, do some action or if it is dark out, turn on a light. Identifying patterns and using structures such as repeats and loops to execute them efficiently.
Representation	Symbolism develops in early childhood. The ability to represent concepts as symbols is important for computational thinking. The formal languages of computer science are representations of the programmers thoughts organized in such a way that a machine can understand them.

Hardware/software	Hardware and software are parts of a system. Software is used to control hardware. Hardware is built to interpret software and do some action. Depending on the hardware, this may be interpreting large data sets (computer) or navigating a maze (robot).
PROCESSES	
Design Process	A cycle with no explicit beginning or end where a child: asks questions, imagines, plans, creates, tests and improves, and shares their work. Engaging with and iterating through these actions is design process
Debugging	A systematic approach to isolating and addressing problems within an existing piece of work. For example one might step through a program to find an error or check all connections on a piece of hardware.

In early childhood computational ways of thinking are also important for the development of other low-tech, non-digital learning. Sequencing, the core of algorithmic thinking, has been the subject of developmental scholarship long before computers. Sequencing is important for children's growing ability to tell stories, put on shoes, numeracy skills, and learning to read. Kazakoff, Sullivan, & Bers (2013) showed how using a robotics curriculum had a positive effect on children's sequencing abilities.

As computers become ubiquitous in today's society so too should these skills in order to best utilize them for enhancing our lives (Wing, 2006)! Barr & Stephenson (2011) and Lee et al. (2011) have outlined the importance of computational thinking in K-12 education. They posit that computational thinking leads to confidence in dealing with complexity, persistence with difficult problems, the ability to handle ambiguity, and the ability to deal with open ended problems.

Computers have greater processing power than humans ever will, this power can only be harnessed if humans can structure the tasks for computers to execute. Ken's second mantra was "save early, save often." At the time my "life's work," some LOGO and Scheme programs, fit on a floppy disk. My 1.44 megabyte floppy disk is a laughable amount of storage in today's world where gigabytes and terabytes are the new standard. However, then and still now the most powerful programs, represented as text, still fit on a floppy disk and are used to operate on and make sense of the gigabytes and terabytes of data we have generated.

Play & Development

Todd, age 4, has been in battle for the last ten minutes defending the moon from aliens, large blocks. His dad is packing his lunch in the other room. In an effort to get him to daycare on time he calls out, "I'm using my blaster now…Boom!" Todd says, "look you got 'em!" and knocks down all the blocks and says "I'm ready to blast off, 5,4,3,2,1!!" Todd's dad enters the playroom, scoops him up and they head out for the day.

Play is an important source of development in early childhood. It provides a safe place for children to make sense of the confusing world around them. It is a time for them to fulfill their unsatiated desires. The ability to symbolize, that will later turn into abstract thought and reasoning, begins in play. It is a place where they will begin to learn to take the perspective of others. (Vygotsky, 1978; Scarlett et. al., 2004)

For Todd the blocks do not stand in the place of aliens, they are aliens. This is early symbolism, to use Vygotsky's (1978) terminology the block is a pivot. Operating pragmatically within his story world the act of knocking them over destroys them. Through this play Todd is exercising skills that are the precursors to full symbolic representations that will one day allow him to do algebra, or write a screenplay.

Educational robots may act as pivots much the same as a stick. They offer the researcher a window into a child's ability to symbolize. They do not necessarily drive symbolic development, however once a child becomes proficient with a robot—which requires the ability to make symbolic representations with software—the opportunities for them to express themselves symbolically far out weigh those of the stick.

Children in early childhood seem to move fluidly between the "real world" and the world of make believe. But there is something more nuanced going on. Children develop a boundary between the real world and their story worlds. They begin tightly fused to their imaginary worlds with little distinction between what is real and what is made up. With development a boundary emerges which they are able to use to their narrative, theatrical, and intellectual advantage. (Scarlett & Wolf, 1979)

There is little research specifically about kindergarten age children engaging in play with large objects. Ware, Uttal, & DeLoache (2010) describe a phenomenon in children ages 1-3 known as scale error in which a child attempts to perform a task that is impossible due to extreme differences in the size of the objects involved (e.g. sitting in a small doll chair). An important distinction about play involving scale errors is that they are not acts of pretense. Bender (1978) describes the importance of using large wooden blocks in preschool (age 4). She describes how large blocks allow children to build enclosed structures for privacy, as vehicles of dramatic play, and for the express purpose of physical activity. Rufo (2012) describes the importance that a set of forts in the woods had to his fourth grade classroom. Over time up to 25 children were engaged in building forts and setting up supply chains to construct them. This required a great amount of ingenuity and communication between the children.

The Rigamajig and Tubelox are two tool kits developed for children to build on a large scale. Their forms require children to work together to accomplish their goals. There is no research on either of these tools however I imagine they offer many of the same opportunities that Bender (1978) describes in her study.

When his father shoots the blaster and Todd cries out "you got 'em!" he has no idea that his father is not seeing what he is seeing. "Thought in the child is ego centric, i.e., … the child thinks for himself without troubling to make himself understood nor to place himself at the other person's point of view"(Campbell, 1976). The most salient example of this phenomenon is the three mountain experiment (Piaget & Inhelder 1956). In which a child sits in front of a table top model of a mountain. There is a doll placed at another position on the table. The child is asked to select from a set of images what the doll is seeing. Children under the age of eight have difficulty with this task, and children under six often choose the picture representing their own point of view (McDonald & Stuart-Hamilton, 2003).

Summary

Play is an important and productive space for children to develop emotionally, intellectually, and physically (Scarlett et al., 2004; Vygotsky, 1978; Curtis & Carter, 2014). Constructionism asserts that powerful learning occurs through the act of creating personally and epistemologically meaningful artifacts (Papert, 1993; Papert & Harel, 1991; Bers, 2008). Many educational technologies, including programing languages and robotics platforms, of various levels of complexity and modularity have been designed in the wake of Papert's (1993) initial work (Blikstein, 2015).

Introducing computational thinking in K-12 promotes children's confidence in dealing with complexity, persistence with difficult problems, the ability to handle ambiguity, and the ability to deal with open-ended problems (Lee et al., 2011; Barr & Stephenson, 2011). Tools such as KIBO (A. Sullivan et al., 2015) have been designed to bring these practices to the early childhood setting.

In 1978 Judith Bender wrote "there is not only a scarcity of professional literature relating directly to large block building…but there is also little emphasis in training teachers regarding the values and use of these blocks in early childhood education." To date this topic, as well as large manipulatives or toys of any sort, is still largely unexplored in the literature. As researchers take on this task its relationship to educational robots and computational thinking should be considered.

Chapter Three: Statement of Problem

There are benefits of robotics in early childhood. Robots can serve as mediums of expressions for all learners (Bers, 2007; Resnick, 2006), they support young children's sequencing skills (Kazakoff et al., 2013), and promote sensorimotor development (Marina Umaschi Bers, 2008). Research shows how children's engagement in fantasy play is both natural and necessary for development (Vygotsky, 1978; Scarlett et al., 2004). The playground provides children with an opportunity to engage in important gross motor play. It is also a world of big objects for children to cast their imaginations on.

Yet there are no products or research focused on combining early childhood robotics with gross motor and large-scale fantasy play. This study is in part motivated by questions like: why do we only provide our children with small desk (or floor) top sized robotics kits? In so doing are we inadvertently constraining their powerful imaginations and budding computational thinking practices to micro-worlds? Can robotic platforms for early childhood support gross motor development (Curtis & Carter, 2014)?

To begin investigating the role that physical scale can play in early childhood robotics I have built a large, rideable, and programmable robot for young children based on the KIBO robotics platform. This exploratory study centers on the research question: How are children's experiences with big-KIBO different from KIBO?

Chapter Four: Developing big-KIBO

Overview

This section begins by describing the KIBO robotics platform and introduces both the original KIBO and big-KIBO. This section also contains two pilot studies that were carried out during the development of big-KIBO. Pilot study 1 was carried out before building big-KIBO to determine its dimensions. Pilot study 2 is a set of the first play sessions with big-KIBO. Pilot study 2 was carried out as an informal usability test to see if and how children could use big-KIBO. It was also designed to test initial ideas for the activities in the final study.

KIBO Robotics Platform

KIBO

KIBO (Figure 1) is a robotics platform specifically designed for children ages 4-7. It consists of a main robotic body with easily attachable parts such as motors, sensors, and a light. KIBO is controlled with a TUI made up of cubic wooden blocks (A. Sullivan et al., 2015)(Figure 2). KIBO has a barcode scanner for loading programs. KIBO has a single triangular shaped button used to turn it on, and to execute or terminate programs. The button has a green light inside that flashes or stays lit depending on the robots state.

Figure 1. KIBO Robot

Each block of the TUI has a command (FORWARD, END, etc.) represented by a symbol, the name of the command, and a barcode printed on four sides on the block. One side of the block has a dowel coming out of it, the other has a hole to accept a dowel—the BEGIN block only has a dowel on its right side and the END block only has a hole on its left side, this prevents children from making syntax errors. Blocks are assembled into “programs” by creating a string of blocks connected by inserting the dowel of one in to the hole of the next. From here on in this paper blocks and their commands will be represented by typing out the command in all capital letters (e.g. BEGIN stands for a green block with a circle on it, the word begin, and a barcode that indicates to the robot that the block begins a program). The REPEAT block is twice as wide as the other blocks and has a velcro patch that accepts a parameter sticker. For this study only the motors, motion blocks, and REPEAT blocks were used.

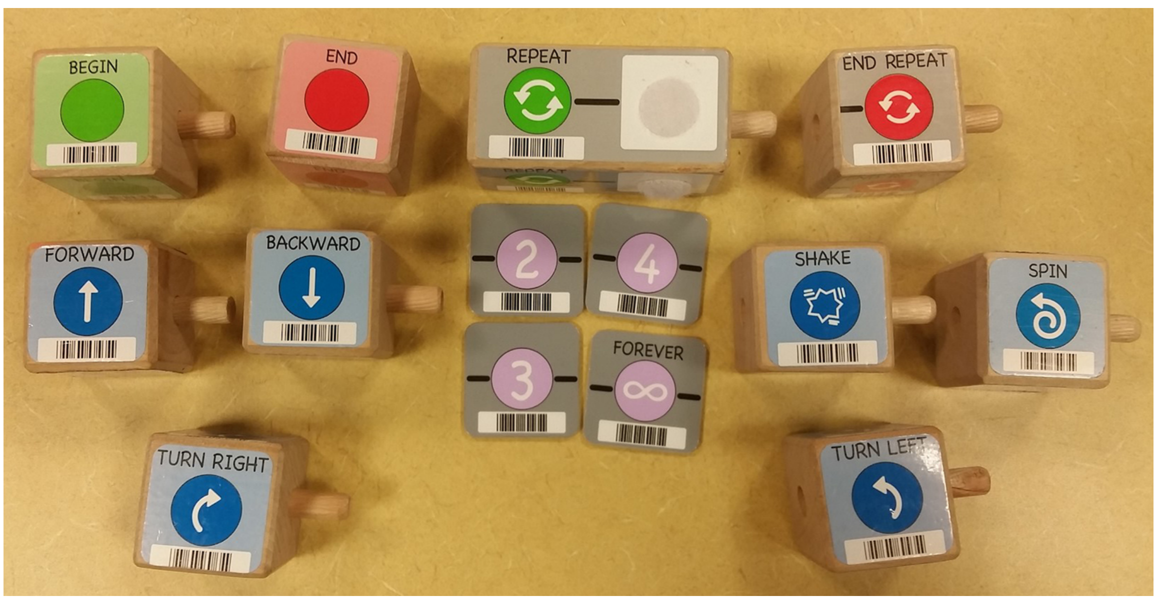

Figure 2: KIBO Blocks Used in This Study

KIBO’s parts are all made from a mixture of smooth plywood and plastic that easily fit in a child’s hand. The wood was chosen to add a natural feel to the robot. The plastic is both a functional piece and a way to provide children with different types of materials. The bottom of KIBO is transparent, allowing the child to see the electronics that control it. This is a great discussion point for children and educators about what makes a robot? and how do they work?

KIBO (Figure 3) is a commercially available product through KinderLab Robotics (“KIBO,” 2014) a complete description of its peripherals can be found on their website. KIBO was developed in the DevTech research lab and lived under the name KIWI while in its prototyping phase.

Big-KIBO

Big-KIBO is a replica of KIBO that is 109 times bigger (by volume) than the original. It was built with the intention of enabling children to control something “big,” and to allow them to ride their creations if they choose. “How big is big?” Is addressed in pilot study 1. It was constructed for this research project. I conceived of and built it as a final project for the class *How to Make (Almost) Anything* at MIT in the Fall semester of 2016. The funding came from the Early Childhood Maker Space (ECMS) project funded by the CEEO Innovation Fund, the LEGO Foundation, and the National Science Foundation.

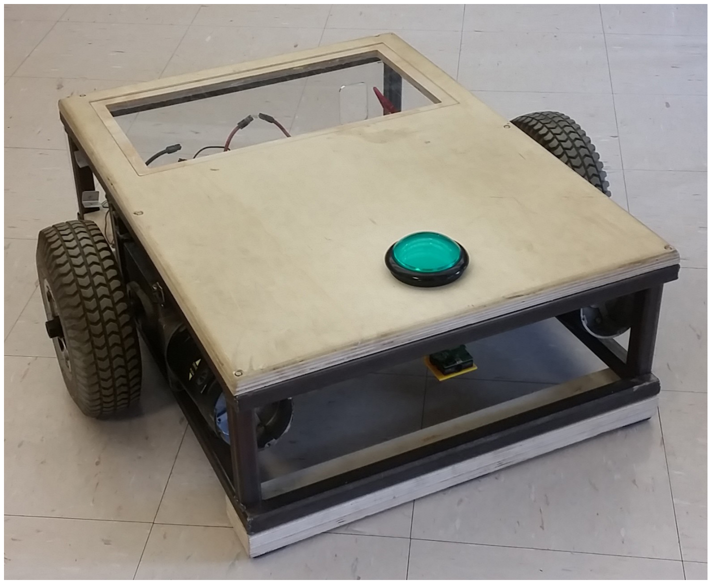

Figure 3. big-KIBO

Due to the nature of the big-KIBO project there are a few differences in the way it functions from KIBO. When KIBO is on its barcode continuously scans. Big-KIBO requires the user to press a small button on the top of the barcode scanner to activate its scanner for each block. Once a program is scanned KIBO "remembers" it. By pressing KIBO's button again the same program will run until a new program is scanned. Big-KIBO does not "remember" its programs. Each time a program is executed it must be rescanned before it can run again.

Big-KIBO's size and weight make it so that the motors and wheel are permanently attached, where as, KIBOs motors and wheel can be added and removed. Big-KIBO cannot be lifted above the blocks, which is a popular scanning method when using KIBO. Users cannot see the bottom of big-KIBO, to preserve the ability to see inside big-KIBO a piece of clear acrylic was inlayed into its wooden top. The plastic sides of KIBO were replaced with steel in big-KIBO to support the weight of the robot and children. Highlights of the design process, plans, and code for big-KIBO can be found online at www.mvizner.com.

Pilot Study 1: How Big is Big?

Objective

In the processes of designing big-KIBO I ran into the question of “how big is big?” I puzzled for a while in my head and in my drawings, and decided the best way to find out would be to ask some children.

Materials

I constructed two wooden models (Figure 4) of how big a big-KIBO could be. The first 22”x30” was designed to fit through the doors of Elliot Pearson while maintaining the same ratio of width to length as original KIBO. The second 36” x 48” was made to be even bigger while maintaining the same ratio. Both models had wheels that would allow it to be maneuvered by the children. The wheels were placed so that the models could turn in place like KIBO.

Figure 4. big-KIBO Prototypes

Subjects

I recruited three 1st graders from the Eliot Pearson Children's School (EPCS), two female and one male.

Procedures

The children were brought to the ECMS by an assistant from EPCS. Both model robots were present in the space when they arrived. After a brief conversation about making things and prototypes I said "I want you guys to get up and play with both these things and tell me if you can figure out what they are for." They were given 10 minutes to play freely with the model robots and some of the other materials in the room. I then explained to them that these were models for a robot like KIBO that I was building. I asked them their opinions about each one. I also asked them how they could ride on each one and how many people could fit on each. They were then given 5 minutes to play freely in the ECMS.

Summary of Observations & Findings of Pilot Study 1.

While playing with the two model robots children spent the majority of their time loading logs, blocks, and each other on to the models and pushing them around the space. Testing the limits of how much each model could hold and how much they were able to push. The destination seemed to be irrelevant except for an important mission in which they stuck their tongues out at the research cameras.

When asked their opinions of the two models their answers pertained mostly to the number of people that could fit and how easy or hard each one was to push.

I noticed that the small robot provided enough space for two children to ride together, or one child and piles of blocks and logs. In my reflections after the study I noted that the larger robot presents as a stage, the smaller is more intimate and affords a relationship with the object (M. Vizner, field notes, 11/18/17). It was this realization that determined the size of the robot, 22” x 30”.

Pilot study 2: Getting Acquainted

Overview

The goal of this second pilot study was to get a few children and their parent’s reactions and opinions of big-KIBO. It also presented an opportunity to pilot some of the activities I used in the final study below. There was no fixed protocol for this pilot. The pilot took place over four sessions. I offer selections of description and findings from each. All of the subjects were recruited as the children of personal friends from the greater Boston area, all names are pseudonyms.

Session 1: big-KIBO comes to life

***Subjects*.**

One six year old boy, Hank.

Observations

After school, Hank's mother brought him to ECMS. Our session begins with me explaining big-KIBO and demonstrating how to scan the blocks. Together we scan the program BEGIN, FORWARD, END. Hank presses the button. The robot jolts forward and Hank yells with excitement while hopping out of its way. The robot stops and he immediately hits the run button again, this time no response. I explain that big-KIBO can't remember and must be reprogrammed each time.

Hank quickly begins to assemble another program. I push the button on the scanner while Hank presents the entire program. I press the scan button while hank slides the blocks over from left to right. Hank climbs on top of the robot in excitement and presses the run button. Nothing happens. I explain that we must have made a mistake while programming. Hank comes back to the bin of blocks—the bin contains all of the extra blocks from the DevTech lab—and makes a bigger program. This time I suggest that he lay the program on the floor, he presents one block at a time, and then places it back in it's place in the sequence. He continues to present one block at a time while I press the button.

The program is loaded, we move some things out of the way and he hops on the robot and wriggles with excitement while I ask him to wait for me to grab my camera. He presses the big green run button, the robot shakes, spins, and carries him backwards as he laughs. I drag the robot back to the center of the room and Hank presses the button again, nothing happens.

Later in the session while Hank is trying to get the big-KIBO to "clean up a mess" he begins to scan the blocks by himself. Pressing the barcode scanner button with one hand and presenting the block with another. A fine motor task I was unsure a six year old would be able to perform.

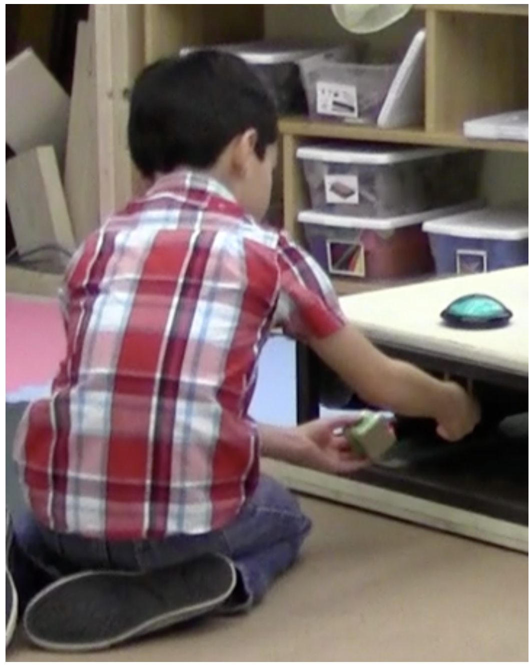

Figure 5. Hank Scans Blocks by Himself

Findings.

In this session with Hank I found that programming and riding a robot is appealing to, at least one, young child. It is possible for a Kindergartner to use big-KIBO's scanner on their own. Hank's repeated pressing of the run button, both when he did not know, and also when he did know the robot wouldn't go made me reflect on the importance of this feature in KIBO. It is in replaying a program that one has an opportunity to debug it.

In Hanks playing with big-KIBO I did not notice anything that he could not have accomplished with KIBO. However he could not physically move big-KIBO, I wonder how the ability to control an object that he could not control with his body made him feel?

Session 2: Maze

Subjects.

A brother and sister, Lucas 6 years and Jessica 4 years.

Observations.

After explanations and free play I explained to Jessica and Lucas that we would build a path for big-KIBO to follow. They built a path by re-arranging the floor tiles in the maker space. The path was like a 'T' with the top line shifted over to the right. Lucas wanted the robot to start at the bottom go to the top, turn to the left, reach the end of the line, and then travel backward until it reached the goal and knocked over a log. He acted this out with his body walking the maze, sticking his arms out, and turning his body to show us what he wanted the robot to do. I went to the office to bring him more blocks so that he could make the program and present it all at once.

Seated with both the robot and the maze in front of him Lucas begins to plan his program. Looking back and forth between the blocks in front of him and the maze ahead he says "I need a FORWARD, [looks for the block, assembles], a FORWARD [looks for the block, assembles], and one more FORWARD [looks for the block, assembles]." He realizes this is the part where he will need to turn the robot. He gets up on his knees and looks at the blocks around him. His father jokingly offers the SHAKE command. So deep in thought he—seemingly without processing what his father had said—says, "ya a shake" he takes it, and then realizes it was a joke. We all laugh and he tosses it back. Then he accepts it anyway and places it in the program. While still looking down at his blocks he says, "[adds LEFT] turn LEFT, [looks up at maze] oh no wait [pulls it off] turn right [looks around for a right block]." His father says, "is it a left or a right?" He thinks for a moment adds the LEFT block back and says “oh ya.” Then he sits up high on his knees and begins to point to the course as he looks at his blocks and iterates through commands by pointing, shifting his wait and turning a bit, presumably to simulate where the robot will face (Figure 2). He says “uh uh [as in no]” sits down to change his program, then gets back up and repeats the pointing task. “oh ya.” Then he completes the program.

Figure 6. Lucas Engages in Body Syntonic Thinking

Findings

When Lucas first explained how he wanted the robot to run the coarse before trying to program it he was very confident in his decision and path planning. I was impressed with his decision to turn once and then complete the maze backwards. When he reached the point in his programing phase where the big-KIBO needed to turn he hesitated, I interpret this hesitation to show that it was not as easy for him to take the perspective of the robot as it was for him to walk the maze and think out loud. This is developmentally appropriate as he is at the age where he is transitioning out of the Piagetian preoperational stage and moving away from ego-centrism (Campbell, 1976). He was able to solve this problem by thinking through it and engaging his body in body syntonic reasoning (Papert, 1993).

Session 3: Imagine later

Subjects.

Sisters Tania, 6 years, and Taylor, 8 years.

Observations.

Tania and Taylor came to the ECMS one afternoon during February vacation. Their mother told me they had already had a long day. I introduced them to both KIBO and big-KIBO. They spent some time playing with the robots and then I explained that we would have a dance party. Taylor spent her time playing with KIBO figuring out how to use the sensors she found in the box. Tania spent her time programing big-KIBO to spin around slowly by moving FORWARD then BACKWARD—this worked because of the slightly out of sync motors which made it turn around over time.

Figure 7. Tania & Taylor Dance on Big-KIBO

The most interesting part of this session occurred after we left the ECMS. I gave Tania, Taylor, and their mother a ride to Davis. In the car they began talking about all the things they could do with big-KIBO including building a fort putting it inside and having it be the guard, and using it to find buried treasure.

Findings

It occurred to me that presenting KIBO and/or big-KIBO is a lot of information for children to absorb at once. Tania and Taylor had an understanding of what these robots were and what they could do, however these ideas were still settling in. It was only after we left the ECMS that they were able to start thinking *with* the robots rather than just *about* them. Incorporating it into their imaginary narratives rather than trying to figure out how to program it.

Session 4: Chinchilla

Subjects.

Lucas 6, and Jessica 4 for their second visit.

Observations.

In this session I asked Lucas and Jessica if they wanted to turn big-KIBO into an animal. After some brainstorming and help from their father they decided to build a robotic chinchilla (Figure 8). After some discussion of what big-KIBO would need to become a chinchilla they set to work. Jessica taped four block legs on to big-KIBOs frame while Lucas added a tail, coating rope in duct tape so that it would be the correct color. Once the tail was complete Lucas was reminded that the chinchilla would need a head. After listening to some suggestions from his father and I he stopped us and said, "I have an idea, you know what, you know if you make a globe you cut a bunch of long ovals and put them together." I offered him to try it in cardboard. He decided that he would need six ovals and traced the first one in cardboard. Together we cut it out using the powered scissors. We continued this process together until there were six nearly 2' roughly identical ovals. Because we were running short on time his father helped us put them together and in the end the chinchilla had a head. Unfortunately there was very little time left for programing so Lucas and Jessica decided to program their chinchilla to SPIN FOREVER.

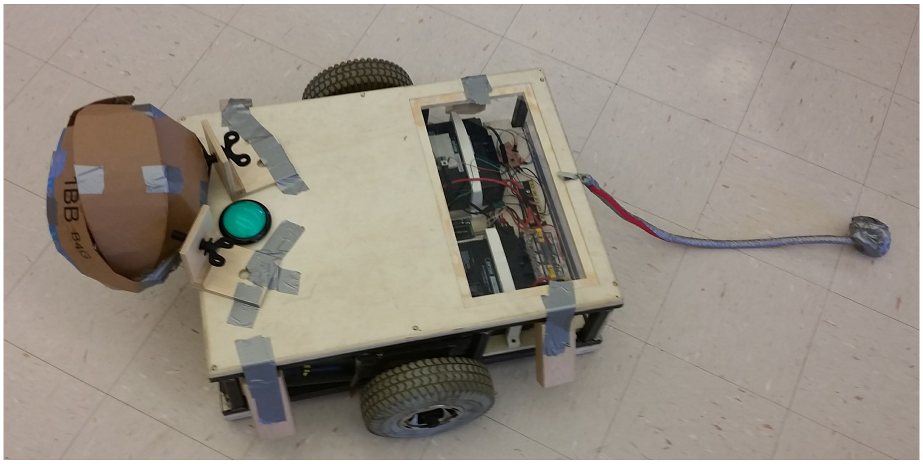

Figure 8. big-KIBO as a Chinchilla

Findings.

A bigger KIBO means bigger decorations. Lucas's chinchilla head was an impressive execution of three-dimensional planning on his part, although he did get some help. The larger format decoration may be a fruitful place for children to explore more complex designs. Perhaps this will lead to an even more varied set of content than can be created for KIBO.

Summary

These pilot studies were an important part of designing big-KIBO, the proposed study, and early themes to investigate. Pilot study 1 was used to determine the dimensions of big-KIBO. I used the same length and width dimensions from the smaller model. Height was adjusted due to material constraints. Pilot study 1 also presented big-KIBO as an object that could be used to transport things other than the user. A theme that I later continued to notice in children's work with big-KIBO.

In pilot study 2 I found that Kindergarten children had the motor capacity to be able to scan the blocks with big-KIBO, a task I previously thought might need facilitation. The users' expectations that big-KIBO would be able to rerun a program after it was scanned was clear and led me to realize it's importance for debugging and learning. I also learned that big-KIBO does not travel straight over long distances. For some children this is overwhelming for others it becomes a feature. I was able to fix this bug before the children in the study reached the maze solving task.

The open ended prompts like program your robot to dance seemed more challenging. Giving children multiple opportunities to engage with big-KIBO as opposed to one time exposure is important for children to be able to begin to think *with* it. For the case of the chinchilla this open ended prompt allowed kids to craft (the chinchilla head) on a much bigger scale.

I observed that using one's body to plan the robots motion is nuanced. It is one thing to imagine oneself as the robot, and another to then imagine the robot as one's self. It is also not trivial for children to convert their motion—in the case of a dance move—into a command that KIBO can understand. Both are related to the idea of body syntonic understanding of the robot as well as perspective taking.

Chapter Five: Methodology

Participants

Six Kindergarten children were recruited from EPCS. At the time of the study they ranged in age from 5 years 4 months to 6 years 2 months. The headmaster of the school helped to identify children who had little to no previous experience with KIBO. Three boys and three girls were selected to take part in the study. All participants were enrolled in extra care time at EPCS and attended the research sessions just after lunch during the time they would other wise be resting. The participants' teacher accompanied them to each session and was an active participant in each session.

The participants were divided into two groups, Group A and Group B. The groups were chosen based on the participants' availability. Having mixed gender groups was also a factor in the grouping. Group A consisted of two boys, Matt and Noah, and one girl, Jen. Group B consisted of one boy, Vinh, and two girls, Hadar and Liana. All names are pseudonyms

Materials

Both KIBO and big-KIBO were used for this study. Children were offered a variety of craft materials (pipe cleaners, construction paper, markers, crayons, tape, scissors, fabric, etc.) as well as a variety of building materials (cardboard, Rigamajig, LEGO, etc.) when working on open ended design tasks.

Design

This research project was designed to compare KIBO and big-KIBO, while taking into account the order of presentation and learning effects. Each group spent three weeks with each robot. Group A spent three weeks with KIBO and then three weeks with big-

KIBO. Group B spent three weeks with big-KIBO and then three weeks with KIBO. A set of three activities, described below, was developed and implemented twice for each group. Each participant participated in each one of the activities with each of the robots. Figure 9 presents the research design.

Figure 9. Research Design

Procedure

Activities

The following three robotics activities were developed based on KIBO specific curriculum from the early childhood robotics network hosted by the DevTech research group at Tufts university ("Early Childhood Robotics Network," n.d.).

Activity 1: Robot Dance Party

- The participants were introduced to the KIBO platform.
- The participants were given some time to play with the robot.
- The participants played KIBO says led by the facilitator (a variation of Simon says)
- The participants danced the hokey pokey together with the facilitator
- The participants were asked to program KIBO to dance the hokey pokey

- The participants test their program and dance the hokey pokey together with KIBO
- The participants were told that the next activity would be a robot dance party.
- Participants shared their favorite dance moves then programmed KIBO to copy them
- Music was played and participants were given time to dance and make more dance programs.

Robot Dance Party was designed to introduce participants to the robot and programing concepts associated with it. It offered both a goal oriented task and an open-ended task. It allowed children to play freely and also have some direction.

Activity 2: Robot Mazes

- Three mazes were made out of floor tiles prior to the participant's arrival (Figure 10). The first was a straight line using three floor tiles. The second was an L shape made from three floor tiles. The third was a square with a hole in the middle made from eight floor tiles.

Figure 10. Mazes Used for Activity 2.

- The participants were asked if they noticed anything different about the space.
- The participants were asked to solve the first maze.
- After solving the first maze participants were asked to solve the second maze

- After solving the second maze the participants were introduced to REPEAT
- The participants were asked to solve the third maze

Robot Mazes was designed with Piaget's three mountain experiment in mind, i.e. I was interested to see if the size of the robot made a difference in the way children take different perspectives. The progression of shape and difficulty of the mazes was designed to scaffold the use of repeat blocks as a control structure.

Activity 3: Free Play

- The participants were given an opportunity to play with their robot however they choose. They were offered a collection of craft materials shown in Figure 11 and the use of anything else in the ECMS to decorate or transform their robot.

Figure 11. Craft Materials Available for Activity 3

- If the participants needed guidance they were directed to think about what type of animal they might want to turn their robot into.

- At the end of the third session the participants were informed that next week they would be playing with a different robot.
- At the beginning of the last (6th) session the participants were informed that both robots would be made available half way through the session.

Free Play was designed to see what participants would transform big-KIBO and KIBO into when given no prompts and the materials available in the ECMS. I chose this as the last activity so that participants would have time to familiarize themselves with the robots and be able to see it as a tool to think with, instead of focusing all of their resources on understanding what it can do. I chose to introduce both robots in the last session to see what children would do when offered both robots at the same time.

Clinical interview

During the activities I engaged the participants in clinical type interviews (Ackermann, 2003) to illicit their thinking. I asked the participants questions about their work like:

- "I see that you are working on ___[Insert objective observation about their work]. Can you tell me why you're working on that?"
- "I'm so interested in this part of your work! Do you remember what you were thinking when you made it?"
- "You've said that you are working on ___[use their own words to describe their task]. Can you tell me how this [use specific language and/or show object you are asking about] will help you complete your project?"
- "Can you tell me about what you are working on?

- "I notice that you have stopped working on ___[if possible, use their own words to describe their project]. Can you tell me why?"

Data collection

Video, researcher notes, and photos were collected as data for this study. Video was recorded on two cameras placed in the early childhood maker space. Researcher notes were take by a research assistant when their schedule allowed. My own reflections were recorded immediately after each session. Photos were taken on a digital camera and iPad.

Chapter Six: Analysis

Overview

In this section I present two tools that were assembled conceptually from literature and my own experience with KIBO, examined using the data collected in this study, and refined into the version presented below. The tools are lenses through which I analyzed my data. I also present observations and an analysis of the ways children played with the robots in their free play sessions.

The first tool (Table 2) is a developmental model of programming with the KIBO platform. This model describes four distinct stages of programming with the KIBO platform. It may be used to locate a child in their trajectory towards fluency with the KIBO platform. Outside of this study it may have implications for curriculum development and best practices for teaching programming in general in early childhood. The second tool (Table 3) is a set of actions that children engage in when using the KIBO platform that reflect the powerful ideas of computation thinking. This table operationalizes the computational thinking framework so that it can be used in the analysis of both KIBO and big-KIBO.

I provide narrative examples to illustrate the developmental model and the actions of computational thinking. I examine the intersection of computational thinking framework and the development of children's ability to program the KIBO platform.

I end this section by presenting moments from the free play activities from both groups to highlight the ways in which children play with KIBO and big-KIBO. I analyze these sessions for differences both in structure and content of play.

Developmental Model

"Development…is a movement towards perfection, as variously as that idea may be constructed" (Kaplan, 1983). In this case perfection is the ability to use the KIBO platform fluently to solve problems and express ideas. In this section I present a developmental model that describes the movement towards a capacity to program KIBO (and/or big-KIBO) with fluency. The model contains four stages. They are proto-programming, early-programming, programming, and fluent-programming. I describe the actions that characterize each stage.

To be "in" a stage it is not required that a child do all of the actions described for that stage rather the predominant behaviors they exhibit are in line with those described for that particular stage. While the model presented below appears—and is in part—linear it should be noted that it is also cumulative, at any stage a child may exhibit actions characteristic of another stage. Some theories (Feldman, 1994) suggest that transitions occur only after serious regressions. Others do away with strictly linear models of development all together (Thelen & Smith, 1998)! Tackling the subject of transitions between stages, while fascinating, is outside the scope of this work.

This model serves as a lens for structuring the development of programming with KIBO. Later in this paper the powerful ideas of computational thinking will be considered together with this model. It also serves as an example for other researchers interested in programming in early childhood as a way to view development.

Proto-Programming.

A child that engages in proto-programing understands that the blocks and the KIBOs are objects to be manipulated. They have the fine and gross motor skills to manipulate the hardware (robots) and software (blocks). Some may see the blocks and robots as having a relationship to each other. However, they do not have a grasp on the parts of the system as means to ends and/or the relationships between the parts of the system.

A proto-programmer plays with blocks as blocks, making chains, stacking them, or throwing them about. The form of the wooden KIBO blocks affords placing the dowels of one block into the holes of the other. A child could “write” a syntactically correct program with out knowing by capping the ends of a chain of motion blocks with BEGINNING and END blocks whose forms terminate the chain. Putting things together and pulling them apart is a common and developmentally appropriate action for children in early childhood (Curtis & Carter, 2014). It is no surprise that on first encounter, and subsequent ones, that children “program.” What then differentiates proto-programing? In subsequent stages a child has an understanding that each block contains some information, an instruction, a bit.

The scanners, and buttons of each robot are enticing features. As objects they signal that they are there to do something. With an early understanding of the relationship between the blocks and the robot children engaged in proto-programming may press, or even force, the blocks into the front of KIBO or big-KIBO, up against their scanner, or into the top near their button.

The KIBO robot “remembers” its last program after it is turned off, which is not a feature of big-KIBO. When a child uses KIBO for the first time it is possible to get a response out of the robot by pushing the flashing green button—pretty sweet deal for a Kindergartner. This is one way a child can come to understand that KIBO is a thing that moves. The child may be compelled to continue pressing the button to get KIBO to move over and over again. Repetition is a typical behavior for children of this age as they come to understand the world (Curtis & Carter, 2014; Richardson, 1998; Piaget, 1961). In this way a child has executed a program with out knowing it was a program.

Early-Programming.

At some point a child develops an understanding of the relationship between the hardware and software of KIBO. the blocks take on their meanings. They become bits that can be combined to control KIBO. Early programs are syntactically correct, starting with BEGIN, followed by motion blocks, and terminated with an END. The goal of these early programs is to make KIBO move.

Two examples of early programming are “the biggest program in the world” and “on the fly programming.” “The biggest program in the world” is a term coined in the DevTech lab. It is when a child uses all of the available KIBO blocks placed between BEGIN and END. If REPEAT or IF blocks are available they may be added to the “biggest program in the world” with incorrect syntax, an indication that they have been added in order to make the program as long as possible. “On the fly programming” is when a child programs a KIBO with out a pre-determined program. They start by scanning a BEGIN block and tossing it aside. Then they pull motion blocks one at a time from a pile, scan them, toss them aside, and repeat until they are satisfied. Finally they scan an END block and run their program, with no record of what the robot will do. “

Early programming techniques such as “on the fly programming” and “the biggest program in the world” demonstrate an understanding of KIBO’s most basic control structure. However they do not use KIBO as a platform that can solve problems or express ideas.

Programming.

The onset of programming is marked by goal-oriented actions. In this stage children realize that they have the power to teach KIBO how to think. Programs in the programming stage are created to accomplish an explicit goal. These goals may be presented to the child as challenges (e.g. program the robot to dance the hokey pokey), or generated internally as ideas they wish to express (e.g. dance moves of their choice).

Programs in this stage all begin with BEGIN and end with END. They typically contain three to six motion blocks. A child may use the REPEAT block with or without the help of an adult. If REPEAT is used, it is without the reproduction of a specific pattern for efficiency. If a child encounters an error in their program they try to build the program again from the start, or replace blocks in their program by trial and error. They may also try to rescan their existing program, with the understanding that there may have been a scanning error.

Programs in the programming stage are solutions to problems (e.g. KIBO follows a pre-determined path) or concrete representations of abstract ideas (e.g. LEFT RIGHT LEFT is KIBO shaking its “butt” as part of a dance sequence). The programs are constructed only with the intention of performing a specific task. No thought is given to the efficiency or elegance of the program itself.

Fluent-Programming.

Fluent-programming occurs when a child has mastered the use of the KIBO platform such that with enough time they can accomplish any goal they set out to.

Additionally they are interested in evaluating their own work to optimize the efficiency of their programs. Fluent-programming differs from programming in the way that a "coder's" work differs from that of a computer scientist.

A program in the fluent-programming stage solves a problem or represents an idea that has six or more steps in it. When fluent-programming a child is reflective of their work. They solve problems and create representations with the least possible resources. They use REPEAT, when applicable, to make their programs efficient.

When a child who fluent-programs encounters a problem with their program they begin the debugging process by stepping through their existing program to find the error. They fix their program instead of starting over. They may also look to other parts of the system as the root of the problem (e.g. perhaps something was missed while scanning, or a motor is in backwards).

Table 2: *Developmental Model of Programming with KIBO Robotics Platform*

Stage	Acts
Proto-programming	• Plays with blocks as blocks not as bits. • Places blocks on top of, in front of, or anywhere near the robot without scanning • Presses run button repeatedly with no response, or pre-programmed response
Early-Programming	• Creates program with BEGIN and END blocks for the sake of making KIBO move. • Creates program with all of the available blocks in order to use all of the available blocks. "Biggest program in the world" • Scans blocks that are not part of a predetermined sequence. "on the fly programming"

	• Creates program that does not solve a challenge or represent an idea
Programming	• Solves 3-6 instruction given challenge without REPEAT. • Represents 3-6 instruction symbolic idea with program. • Uses REPEAT with help from adult and/or without the intention of reproducing a pattern for the sake of efficiency. • Debugs program by trial and error or by creating an entirely new program.
"Fluent"-programming	• Solves 6+ instruction maze with REPEAT when appropriate. • Represents 6+ part symbolic idea with program. • Debugs program by stepping through it. • Uses existing program as a starting point for new program when encountering a bug.

Computational Thinking Framework

Powerful ideas are "personally useful, interconnected with other disciplines, and [have] roots in intuitive knowledge that a child has internalized over a long period of time." According to Bers's (2018) there are seven powerful ideas of computational thinking. Five are conceptual: algorithms, modularity, control structure, representation, hardware/ software. Two are process based: design process and debugging. The KIBO robotics platform is fertile ground for these concepts and processes to take root. In observing children playing with KIBO there are specific actions that are indicative of these powerful ideas.

In this section I present a table (Table 3) linking individual powerful ideas of computational thinking, as presented by Bers (2018), with children's actions with the KIBO robotics platform. This table is to be used for better understanding how KIBO can foster these powerful ideas. Each of these concepts and processes looks different at different stages of a child's development. This table offers the entire range of actions that represent any one powerful idea. A powerful idea is represented in a child's actions if they do one or more of the actions in any particular category.

Table 3: *Computational Thinking with KIBO Robotics Platform*

Powerful Idea	Actions with KIBO
CONCEPTS	
Algorithms	• Creates program to accomplish an explicit goal • Assembles complete program before beginning to scan. No "on the fly programming" • Keeps program assembled during and after scanning. • Builds program one step at a time.
Modularity	• Breaks problems into smaller pieces • Uses their own or other's bodies to decompose task (body syntonic thinking/ program the teacher) • Uses REPEAT to repeat patterns in order to make program more efficient. • Reuses solutions in new problems • May use REPEAT or other blocks without full understanding of how they work, as long as syntax

	is correct.
Control structure	• Uses BEGIN and END appropriately • Presses run button to make robot go • Does not press run button when KIBO is not programmed. • Uses repeat loops with correct syntax
Representation	• Uses blocks as bits, objects that hold meaning. • Select block for function based on color • Program KIBO in order to “teach” it what do.
Hardware/software	• Use blocks to create syntactically correct program that can make the robot move • Use scanner to program robot • Press run button to create motion once the KIBO has been programmed • Describe the visible electronics as the controller, brain, etc.
PROCESSES	
Design Process	• Tests program • Improves program or robot based on test • Asks for help from friends or adults • Shares challenges and accomplishments with friends • Return to same project over and over again during one or multiple sessions • Make changes to program or/and KIBO without starting over.

Debugging	• Iterate through program to find bugs • Recognizes that a bug could come from a missed scan or a double a scan. • Sees bugs as things to be fixed as opposed to failures • Makes changes to program without starting over.

Computational Thinking x Developmental Model

Computational thinking and programming are intertwined. As children develop as programmers so too do their computational thinking abilities. While a complete developmental model of each of the powerful ideas of computational thinking is outside the scope of this work, their relationship to stages of programming is valuable for a more thorough analysis of children's work and play with the KIBO robotics platform, big or small.

In this section I present a table (Table 4) that organizes acts of computational thinking with the developmental stages of programming. Each cell contains a letter representing examples from the data. A key with titles is presented at the bottom of the table. The examples are presented following the table.

Table 4: *Computational Thinking x Developmental Model*

	Proto-Programming	Early-Programming	Programming	Fluent-Programming
Algorithm			D,	
Modularity			B, D	
Control Structure		A	D, C	
Representation		A	D	
Hardware/Software		A	B, D	
Design Process		C	C, B, D	
Debugging				D

A = Vinh's biggest program in the world.
B = Hadar's END Hack
C= Liana plays with infinity
D = Matt and Jen solve Mazes with KIBO

A: Vinh's biggest program in the world.

I notice Vinh on the other side of the room from the big-KIBO working on a program by himself. I go over to see what he is up to. Below is a transcript of our conversation. At times Vinh works by himself while the teacher and I help other children with the robot.

Me: What are you making?
Vinh: I am making a long program, I am making a long program
Liana: ooh make a long long
Vinh: Making a long long program
Vinh: [Continues to add block, talks to himself] turn right, go back, yes a long long program.
Vinh: [Giggling] Look look how long my program, its funny
Teacher: Whose funny?
Vinh: the program [pointing] its so funny. KIBO!
Vinh: [Pulls more blocks out] Shake shake
Me: Vinh how many instructions is that?
Vinh: HAHAHA I'm going to make a long program [throws hands in the air]
Me: Can you tell me how many parts there are
Vinh: [counting 1,2,3…] 25 blocks!
Vinh: [Continues to add blocks to his program]
Me: Vinh I think the biggest program KIBO can remember is 36 including BEGIN and END.
Teacher: How many pieces do you have now?
Vinh: [adds another block, jumps up] now there is a hundred pieces. [Sings] 100 pieces, 100 pieces. (There are less than 100 pieces)

Me: Vinh when I made big-KIBO I didn't think that anybody would ever make a program this big [I begin to count the blocks].
Vinh: Hey stop counting it
Liana: [Looks over at the empty block bin. Surprised] He used up all the blocks!
Teacher: How many is it? Is it 50?
Vinh: Maayyyyybe
Teacher: So let's count, lets see If I was right or not.
Vinh: [crawls to the end of his program and begins to count] 1,2,3…
Me: [run to get my computer in order to check big-KIBOs source code to be able to tell Vinh how long his program can be]
Vinh: 33! [gets up and dances around]

The others have just finished scanning a program for big-KIBO to do a "happy dance. Vinh goes over to the crafting table and puts the dress he made on big-KIBO. While the two other children are choosing a song Vinh goes over to their program and takes off three blocks, counting as he takes them.

Vinh: 34, 35, 36 [he takes these blocks and adds them on to the beginning of his program].

Vinh waits patiently while the girls run their happy dance program two times. When it is his turn he begins to scan. I ask him to keep the program together. He scans each block and hands them to the teacher to keep them in order for him. While he is scanning I check the source code for big-KIBO and find that the maximum number of commands it will accept is 32.

Me: Ok, the most it can take is 32 commands including BEGIN and END.
Teacher: So they can only do 32
Me:[In a lower tone] actually 33
Teacher: So you have to take three out.
Me: because it includes 0.
Me: Vinh I don't know what's going to happen if you do more than 32. Actually I don't think it will go.
Teacher: Ya, Better not because he is doing it so much, because he is doing it so much and then it won't go and then…
Me: Ya, let's say 32.
Teacher: Vinh you have to take 4 out.
Vinh: [angrily] uuuggghhh, Stop reminding me!

Teacher: you know why, because if you're scanning all 36 and KIBO can only do 32 it's not going to move.
Vinh: Ok I'll do 32.
Teacher: Do you want to remove the 4 or do you want Miki to do it for you?
Vinh: I want to remove the 4 [crawls over and removes last 4 blocks in program, and leaves the END].

Vinh finishes scanning. All three of the children position themselves on the robot. Vinh presses the button and giggles as they ride big-KIBO for the longest program ever made.

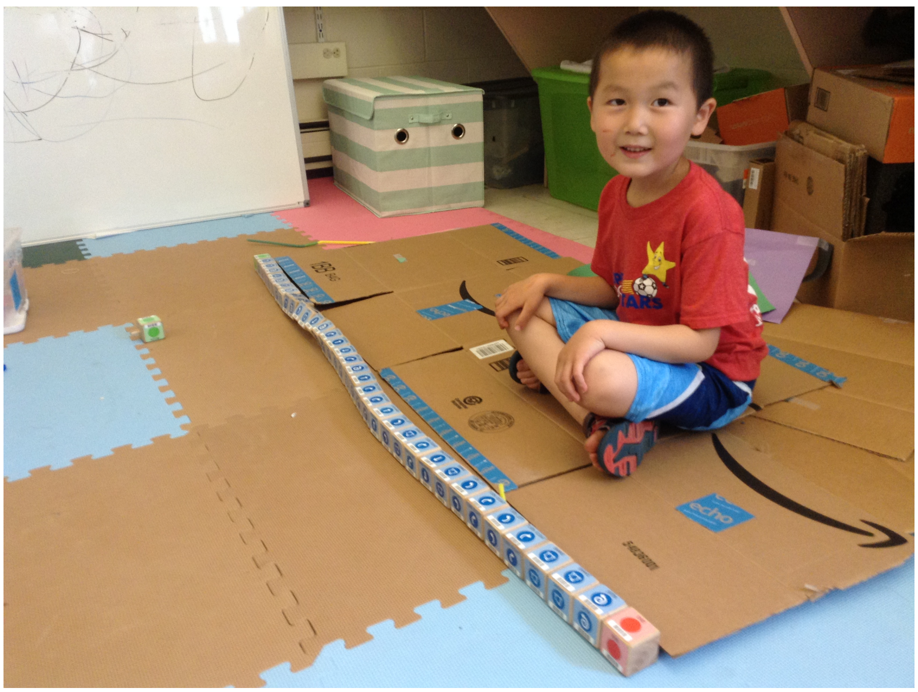

Figure 12. Vinh's Biggest Program in the World

This is a strong example of early-programming. The program that Vinh is creating has no functional goal for the robot, and not meant to represent any action. Vinh makes it very clear that goal of his program is for it to be as long as possible. Along with constantly repeating that he is making a long program his actions also point to his intentions. When he takes exactly 3 more blocks from Hadar's happy dance program counting up to 36 he is maxing out the number of blocks he has been told are allowed.

When I later tell him he has to remove 4 blocks he gets a bit upset and says "stop reminding me," he knows the maximum and is thinking about it.

While building his long program Vinh is also engaging in computational thinking, specifically representation and hardware/ software. This program is being made for big-KIBO. In limiting himself to the maximum number of blocks Vinh shows an understanding that these blocks represent commands. It also shows his understanding the big-KIBO is a piece of hardware with limitations, and even though there is a surplus of blocks (the software) there is a maximum number of commands big-KIBO (the hardware) can handle.

B: Hadar's END hack

During her first session with big-KIBO Hadar discovered a bug in its firmware. After a program is scanned and executed it can be rerun if the user scans the END block and presses the run button. As part of getting to know the robot the group had already made a program, scanned it, and executed it. The program remained in tact and on top of the robot. I then spent a few minutes presenting the next task, hokey pokey.

> Me: How can we get the robot to do the hokey pokey?
> Hadar: ok I'll press the button [presses the run button repeatedly]
> Liana: [lays across the back of the robot holding her cat doll]
> Me: How can we put our robot in?
> Hadar: [takes the end block from the last program] I'm going to do a scan [begins to scan END]
> Me: What's it going to do if you scan that block?
> Hadar: Nothing [completes scan, presses run button]
> [big-KIBO runs the same program they had previously scanned]
> Me: [very confused, trying to hold the robot in place in case it doesn't stop (a previous malfunction)]
> Hadar & Liana: [jump on and off to ride big-KIBO]

A week later the group returns for their second session. Their first task is to solve a maze that is a straight line. Their first attempt contained 5 FORWARD blocks and caused the robot to run into the wall. They decide to try a program with 4 FORWARD blocks. Hadar assembles the program and brings it over to big-KIBO. She scans END first, laughs, and presses the run button. Nothing happens and this time she is surprised. 10 minutes later Hadar is working on a program to solve the right turn maze. She assembles a program and before scanning it she scans only the END block. This time I hit the hard reset button before she has a chance to run it.

Hadar repeats this behavior several times over the course of the three weeks they spend with big-KIBO. Although her discovery of this bug was accidental her re-application of it is a strong example of modularity. Not necessarily in the decomposition sense, but, in the way that one can apply something used from a previous problem to a new one, even if *how* it works is not fully understood. It also shows an understanding for the way the hardware is controlled by the software, even if its not the way it is "supposed" to work. After her initial discovery of this bug her repeated testing, and testing under different conditions is an engagement in the design process. These behaviors point at an effort to understand this phenomenon placing her actions in the programming stage.

In the sixth week when big-KIBO was reintroduced there was a moment when she wanted to run a program again and chose to rescan it, even after the teacher tried to remind her of the hack. Perhaps she forgot, an indication that the developmental model is not strictly linear, or perhaps she had been discouraged by my attempts to stop her from using this hack in the past.

C: Liana plays with infinity

Towards the end of the third session with big-KIBO—in which big-KIBO is transformed into Kiki, a cat, (discussed below)—I announce that there is time for one more program. While crouching on big-KIBO Liana yells, "infinity!" Their teacher responds with, "you do infinity, how about you go do it, Liana go do infinity." Liana gets off of big-KIBO, picks up the REPEAT block, which has the 3 parameter attached to it (the 3 parameter is a square plastic piece with the number three and a barcode printed on it, the back has velcro which allows the user to attach it to the REPEAT block it is shown in Figure 2), and brings it over to where the last program was left on the floor.

> Liana: I want to go infinity two, I mean three times.
> Teacher: Infinity is 3 times? Ok.
> Hadar: [from on top of the robot] Hey infinity times
> Liana: [makes the program BEGIN, BACKWARDS, REPEAT(3) FORWARD, SPIN, SHAKE, END]
> Me: how does it know when to stop repeating?
> Me: you are missing one more block.
> Liana: What?
> Me: There is REPEAT, but no END REPEAT
> Teacher: It will not go
> Me: I think it is over there [point in direction of END REPEAT]
> L: [retrieves END REPEAT] Does it go next to [referring to REPEAT block]?
> Me: Do you remember the sandwich? It is whatever is in between. Whatever you put in between here is what will be repeated [pointing to REPEAT and END REPEAT].
> Liana: [puts END REPEAT in the sequence to the right of the REPEAT]
> Me: So now it says REPEAT, nothing END REPEAT, go FORWARD
> Liana: Nothing here!
> Me: whatever is between here and here [pointing to REPEAT and END REPEAT]
> Liana: [Moves the BACKWARDS from the start of the program to be inside the REPEAT "sandwich"]
> Me: So now it will go backwards three times, and then forward and then spin and then shake one time
> Liana: [whining] Hey can anybody help me.
> Hadar: [rips apart the program]
> Liana: [gets up and walks away]
> Teacher: Hadar, you pulled apart the program, can you put it back in?
> Liana: you're putting it the wrong way [walks further away]
> Me: Liana did you want it to go backwards three times or did you want it to do all these things three times?

Liana: I want it to spin infinity times
Teacher: Come do it again, fix your program
[Liana returns to the work area]
Me: do you want it to spin many times?
Liana: mmhhmm [in agreement]
Me: Then SPIN needs to be where BACKWARDS is [I point to BACKWARDS]
Liana: [switches SPIN and BACKWARDS position's in the program]
Hadar: [removes the 3 parameter from the REPEAT block and replaces it with infinity]
Liana:[looks up angrily from her work] I don't like that!
Hadar: I want it to be infinity
Me: did you want forever? Or did you want three times?
Hadar: [whispers] forever
Liana: [whispers] forever
Me: So if you do this it will only spin, because it will say REPEAT forever SPIN END REPEAT, REPEAT forever SPIN END REPEAT, and it will never get to go FORWARDS or BACKWARDS
Liana: I want that! I want that! I want that! [hops on big-KIBO]

At the beginning of this episode Liana seems to have confused the word infinity with the REPEAT function. This is made clear when she says, "I want to do infinity 3 times." Her goal at first seems to be to use the REPEAT function. This is an act which coincides with the programming stage, in that there is a clearly defined goal (using REPEAT), and that she needs help managing the syntax of the REPEAT function.

This episode could be described as Liana playing with control structure, repeat functions, and the concept of infinity. Her shouting "infinity!" shows that there is something fun and exciting about using REPEAT and/or infinity. It becomes clear that, at least part of, her goal is to spin around on big-KIBO many times, perhaps forever. From her fumbling with syntax and misunderstandings between REPEAT and infinity it is clear that she does not have a complete understanding of the concepts, however she is able to use them to get what she wants and have fun.

The notion of infinity is quite complex, and its mathematical representation and application even more so. Yet here we see children applying it to a robotic cat they intend to ride. In *Mindstorms* Papert (1993) describes this phenomenon of being able to play with and use complex concepts such as randomness in LOGO with out being able to express them mathematically. Here we see this phenomenon playing out with big-KIBO. If Liana can master the use of REPEAT and infinity with Kiki she will be better prepared to apply these concepts in the future and to handle them conceptually when the time comes.

D: Group A solve mazes with KIBO

In their second session with KIBO Jen and Matt (Noah was not present for this session) worked together to solve three mazes, described above, using KIBO. Here I present an excerpt from each of the three mazes they worked on.

First Maze

After having the maze explained a few times Matt connects a BEGIN and a FORWARD block, he reaches for the END block and says:

Matt: Let's do END
[stops himself]
Matt: and then forward again, we need two blocks
Jen: We need more forwards, actually we need more two forwards
Me: How many do you think you need?
Matt: I think two
Me: Two more or two total?
[Jen points to and appears to count the tiles in the maze]
Jen: Three more
Matt: Two total
Me: Ok well you can take out of the box [point to box of blocks], there are plenty of forwards in there, do you want to bring it over here?
[They crawl over to the box]
Matt: I think one total and then we are going to be good
[They dig through the box]
Matt: Oh I found one
[They crawl back to the program they were building and add the FORWARD to their program]
Me: Do you want to test it and see if it works?
[Jen brings the robot to where they were programming]

Matt: I'll move the blocks while you scan.

Matt and Jen work together to scan the program. Jen carries KIBO to the start of the maze and hits the run button. KIBO moves about half way across the maze

Teacher: ooh
Me: good job
Jen: We need more
Teacher: How many more you think?
Jen: I think two
Matt: No, I think we need one

They crawl back to the box of blocks. Jen finds a FORWARD and adds it to the program. They scan the blocks together. Jen puts KIBO at the start of the maze and hits run. This time KIBO stops on the front edge of the last floor tile.

Matt: Nope, we do need four
Jen: One more

Jen returns to the box of blocks and finds another FORWARD. She adds it to the program and begins scanning. Matt announces that he wants KIBO to chase him and stands on the maze. Once Jen finishes scanning she places KIBO at the start of the maze and hits the run button. Matt runs and jumps on a bean bag. This time KIBO makes it to then end of the maze.

Second Maze.

I introduce the second maze which requires the robot to move forward, make a right turn, and then move forward again. Matt says, "oh I know, I think I know". He removes the end block and one of the FORWARD from the previous program. He sits up and points at the maze and asks:

Figure 13. Matt Plans His Program for the Second Maze

Matt: is this way left? (Figure 13)
Teacher: read it, read it
Matt: right, we need to turn left
Matt: [adding blocks as he speaks] turn left, go forward.
Matt: I think I did it wrong. No I think ya.
Me & Teacher: you can try it.
Me: that's the best thing about KIBO, you can try it and if it works great, and if not you can try something different.
Matt: Here we go

Jen scans the program for Matt. They run it. As soon as it finishes Jen picks it up and says "we need more straight pieces." She immediately removes the right turn and looks in the box for another piece. I offer her a RIGHT block and ask if it's what she is looking for. She puts it in the program where the LEFT block was, rescans the program and tests it.

Third Maze.

After several iterations Jen has built a program that gets KIBO to travel in a square shape. Her program is BEGING, REPEAT(4), FORWARD, FORWARD, LEFT, END REPEAT, END. However, when she tests it it passes through the brown square in the center of the maze. When I ask her if she would like to change her program she adds a

FORWARD block as the first block inside the repeat loop. She rescans and when she tests it KIBO stays on the path!

Maze Analysis.

All seven of the powerful ideas of computational thinking are present in this example. In the first maze Matt and Jen go through several cycles of the design process, making increasingly educated guesses based on their trials as to how many forwards it will take to get to the other end of the line. In the second maze Jen's actions could be characterized as fluent-programming she is able to immediately isolate and fix the left /right error they have in their program.

By adding the extra FORWARD inside of the REPEAT block Jen is increasing the length of all four sides of the square simultaneously. What is inside the REPEAT block is a module applied four times to build a square. The REPEAT block control structure allow Jen to do this efficiently.

In each of the mazes an understanding of what the blocks represents and their relationship to the hardware is clear based on the ability of Matt and Jen to use them effectively to solve the mazes. With the exception of Jen debugging actions in the second maze all of the programming done in these mazes is characteristic of the programming stage.

Hadar, Liana, and Vinh use very similar strategies to solve these same mazes with big-KIBO. They also show the application of all seven of the powerful ideas of computational thinking and program at the programming stage. Excerpts from their session are presented in Appendix A.

CT and Development not Captured in Table 4

Not all moments of CT or Programing are captured in this chart. Given the nature of the study there were no strongly representative cases of fluent-programming found in the data set. Given the children's age and the amount of time, and specific tasks they were given perhaps there was not enough space for this development to happen.

The proto-programming column remains empty because computational thinking does not happen during this stage. For a child who programs at this stage the blocks are just blocks they are not yet representations of explicit commands. They cannot be organized into meaningful sequences (algorithmic thinking), or used to intentionally control anything (control structures). Due to the way that I scaffolded the activities—I presented them with partially made programs the first time I introduced the blocks, and showed them how to scan—for the children there are no clear examples of proto-programming in my data set. There is one moment where Liana builds a program vertically, as if to make a tower, she says each block out loud as she stacks, but the order is not explicitly related to the task at hand. While this is more representative of early-programming the tower form is reminiscent of proto-programming actions (Figure 14), because KIBO blocks are used to make a structure that is traditionally made from regular blocks whose structure does not support the act of programming.

Figure 14. Liana Proto-Programming

Also not shown in this chart, but still very important, are the moments of "low-tech" computational thinking that occur when building on the KIBO and big-KIBO. In their third session with KIBO each of the children in group A were working on their own features to add to it. While working Matt articulated an important aspect of modularization.

> Matt: First I am going to get every piece, then, then. How about this you first you get every piece then you decorate it.
> Me: Every piece out of the LEGOs?
> Matt: No no no no no. Like every piece you need, then build.

For Matt it is clear that his end goal will contains sub-parts. He realizes that to ease his work flow it would be beneficial to have all the parts laid out first. This *mise en place* approach is an example of modularity applied by top engineers and chefs alike; the work needn't be "high-tech" to benefit from computational thinking.

The addition of decorations and features to KIBO and big-KIBO engaged the children in the design process at times where programming was not happening. In their final session with big-KIBO Noah decided he wanted to add a "wingbat" to the front of big-KIBO (Figure 15). The "wingbat" is a wooden pulley, taken from the Rigamajig set, attached vertically to the front of big-KIBO. A rope is draped over the pulley and attached to a wooden disk. Noah pulls the rope back and forth to operate the "wingbat"

The pulley is held in a wooden housing. The other end of the housing is round and has a hole in it made to accept a wing nut and bolt from the Rigamajig kit. Noah begins to attach the pulley by inserting a pipe cleaner through the hole and attempting to tape down the ends of the pipe cleaner to the top of big-KIBO. I rip tape for him while he applies it on and around the pipe cleaner. He says, "it's taking me a long time to attach my wingbat." While he works he continuously looks over at Jen's work. Jen has taped a different piece from the Rigamajig kit to big-KIBO. Her piece is two small wooden boards attached at right angles with a single hole in each board meant to accept the wing nut and bolt from the kit. She is making a chair with a seat belt for a small figurine. Noah works for a few more minutes. At one point he lets go of the "wingbat" and it falls over.

> Noah: Uhh I don't think this will work.
> Me: Do you want to try the other idea
> Me: Do you see how Jen put that thing on there [pointing at Jen's work]?
> Me: If you do that then you could screw that into the same piece
> Noah: Ok
> Me: Do you understand what I am saying
> Noah: mhhmm [shakes head]
> Me: do you want Jen to help you?
> Noah: Sophia will you help me make it like yours
> Jen: [shows how she put tape on each side of the board that was attached to big-KIBO]

Noah brings over two of the right angle boards. Sophia demonstrates how to attach one of them. Sophia and I help Noah to set up the board for his "wingbat." As Noah gets ready to attach the pulley he says, "this could be a testing station, Lets make a testing station on

KIBO." When I ask him more about the testing station he doesn't answer and returns to attaching the pulley.

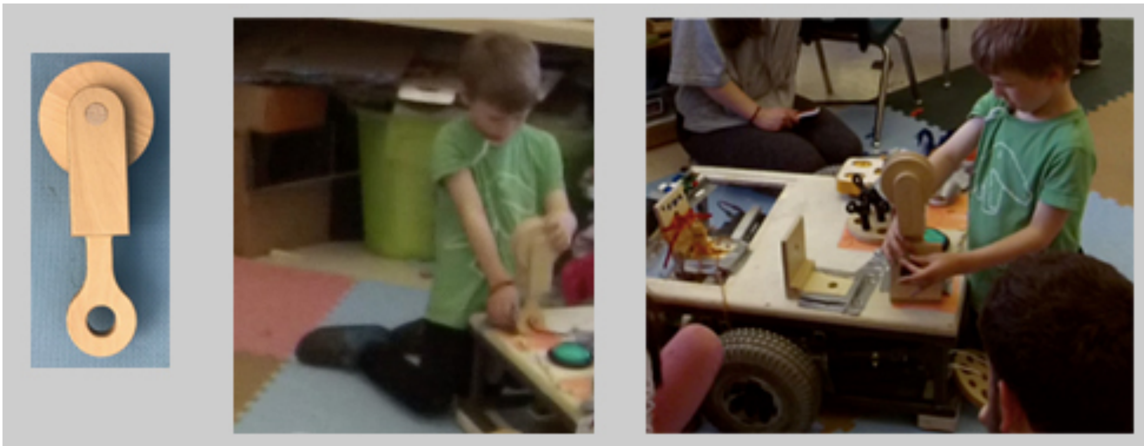

Figure 15. Noah Works on His "Wingbat"

Here we see Noah and Jen engaging in design process. Noah begins by imagining his "wingbat." He then makes the plan of passing the pipe cleaner through the hole and taping it down. He starts to make it, testing his idea. He finds the errors in his design; the tape won't hold it. He shares it with me and Jen. Then through collaboration, and my facilitation, he iterates on his design. He re-engages in the design process of imagining (how Jen's work could help him), planning, creating (attaching the new piece), testing (attaching the string and disk), and sharing.

Play Analysis

"Fantasy play is the glue that binds together all other pursuits" (Paley, 2004), by structuring curriculum to integrate play children fuse their learning of the subject at hand with their own interest and concerns allowing for a richer understanding of both. In this section I present vignettes from each of the free play activities.

Group B Free Play with big-KIBO

For the first two sessions with big-KIBO Hadar and Liana each came with their own cat dolls. The first time they ran a program with big-KIBO it was the dolls that rode on the robot. While they both had cat dolls it was clear that Hadar had a much stronger infatuation with cats. When they arrived for their third session with big-KIBO neither of them had brought their dolls, however, Hadar was wearing cat slippers and a cat ear headband.

I asked the group what they wanted to turn big-KIBO into. After a bit of coaxing from Hadar and Liana they all agreed on a kitty. They quickly named it Kiki, gave it a birthday, and debated how old it was. They each set out to make a different feature of Kiki. Vinh began by making a tail out of pipe cleaner. Hadar and Liana began making pipe cleaner ears, and Liana added decorations—a piece of pink construction paper—over the electronics. They negotiated where the eyes and mouth should go, making fur by cutting the edges off of pipe cleaners, and adding various articles of clothing. Throughout their time with Kiki they spoke to and about big-KIBO as Kiki saying things like “You need clothes,” and “Kiki is crying” (while waiting for someone to program it).

Figure 16. Kiki

Vinh made a heart for Kiki out of a pipe cleaner and taped it on top of the robot towards the back. Later he added two more pipe cleaner rings to the heart. When asked what they were for he said, “for holding on when you sit” while demonstrating. They were handlebars for riding the robot (Figure 17). After demonstrating the handlebars he used the last few minute of decorating time to make a new heart for Kiki.

Figure 17. Vinh Demonstrating His Handle Bars

When it came time to program Kiki I asked the group:

Me: What are some things that cats or cat robots do?
Hadar and Liana: They climb, they crawl, they play with yarn, they eat fish.
Me: Do you think we could program KIKI to eat fish?
Vinh: yes

Liana: I forgot to make fish.
Hadar: [Jumps up and runs over to craft table]
Teacher: Don't make fish, don't make fish. Sit down.
Me: Do you think that KIKI could chase yarn?
All: Ya!
Me: If I give you some yarn could you program KIKI to chase the yarn?
Hadar: Also rotten fish
Me: Which one of those things would you like to program KIKI to do first? to crawl to eat fish..
Hadar: Eat fish! Rotten fish that is real
Me: How could you make a program for KIKI to eat fish, what would that look like?
Teacher: what actions would it do?
[pause]
Me: What action would that be to eat fish
[pause]
Liana: We could use backwards.
Hadar: [Crawls over to Kiki and touches the decorative eyes they added]
Teacher: Will KIKI have to go hunt for fish?
All: yes
Liana: I need to make the fish jump up
Hadar: [Jumps up and runs over to craft table]
Teacher: Wait no no, don't make the fish, what you're thinking right now is what Kiki will be doing
[pause]
Me: First why don't we start programming Kiki to crawl like a cat...
Vinh: Crawl, OK!
Me: What does it look like for Kiki to crawl.
Liana: I know I know. I want to use infinity.

When I asked how their program was like a cat crawling Liana said, "because it's moving around," however each block did not have a specific representation. At the end of their session Liana asked if the robot could remain as Kiki. I explained to them that others would have to use it and that they would be playing with a new robot next week.

Group B Free Play with KIBO & big-KIBO

Three weeks later the same group was given an opportunity to turn KIBO into whatever they wanted. I also told them that after twenty minutes I would bring back big-KIBO. They chose to turn KIBO into a Kitty. The only addition to the robot was a set of ears made from a single pipe cleaner. Vinh then spent time making a toy and a dress for big-KIBO. Liana built a path for both KIBOs to follow. Hadar programed KIBO to do a happy dance. She did not save the program. Later she made another happy dance for big-KIBO. At one point she put KIBO on its side and upside down while running the "happy dance" to see what it would do, pushing the limits of what the hardware is physically capable of.

Figure 18. Hadar Tests KIBO's Physical Limits

When I brought big-KIBO out all three of the children ran out of the ECMS to greet it. Liana and Hadar immediately put ears on it transforming it back into Kiki. Liana put the "blanket" on it she had made, and Hadar programmed it to do a happy dance. When the blanket needed to be removed in order to operate big-KIBO safely she tossed it on the table unintentionally covering KIBO. It was at the end of this session that Vinh executed the biggest program in the world described above.

Group A Free Play with KIBO

In their third session with KIBO Jen, Matt, and Noah worked in parallel each designing and adding their own features to KIBO. Both Noah and Matt tried several designs, not all of which had explicit functions. Towards the beginning of the session the children were told that they could each create their own KIBO creations, as the session went on they decided to make one KIBO together. Here I present the making of their final group design.

Noah cuts a piece of an orange pipe cleaner and tapes it to the side of KIBO. "I'm making antenna," he says. Next he cuts a second piece and holds it next to the first one he cut, and says, "it's too short." He takes the stock pipe cleaner and holds it up to the piece already attached to KIBO, and marks the place he wants to cut with his fingers. I cut it for him and he works to attach it.

Meanwhile Matt has connected four narrow LEGO pieces together in a line with three layers to hold it together. He holds it up and says:

> Matt: you see, now KIBO is going to have a gun. [He places it on KIBO where he plans to attach it then takes it off]
> Teacher: Are you sharing the KIBO right now with him? Or are you going to build one and then after he finishes his.
> Matt: Actually I'm going to share it.
> Teacher: Did you ask him?
> Matt: Do you want to share it?
> Noah: Mhhmm [agreeing]
> Matt: Because sharing is nice.
> …
> [Matt tapes the gun to the center of KIBO]
> Noah: Ooh KIBO has a gun
> Matt: Ya KIBO has a gun
> Noah: KIBO has a shooting gun
> Noah: KIBO has a gun pew pew, KIBO has a gun Pew pew
> Matt: KIBO has a gun pew [teacher puts her hand on his back to calm him down]
> Me: Are you going to program KIBO to shoot the gun.
> Matt: No it's just for decoration

While Matt and Noah were working on the antennas and gun Jen had cut two pieces of pink paper and drawn hearts on them. Once the gun is attached she takes the tape dispenser to her side of the table and begins taping the pink strips to the front and left side of the top of the KIBO. Matt decides to add wheels to cover KIBOs sensor ports.

While he works on this Jen prepares another strip to put on the right side of the top of KIBO. While they work Matt explains.

> Matt: This is quite a decorating KIBO!
> Me: What do you call it?
> Matt: [mumbles then says] Creativity KIBO!
> Me: What can this creativity KIBO do?
> Matt: It can actually do the same thing as the KIBO

Jen places the last piece of paper on KIBO Matt tapes down a small LEGO wheel towards the front of the robot and says to no one in particular, "For the final touch. You see the KIBO, it's a decorated KIBO. Are we going to clean the KIBO up? Actually maybe we could just leave it decorated and then more kids will decorate it even more."

Figure 19. Members of Group A Work On Their Own Features.

Noah and Matt stop building on KIBO and take turns programming it. I ask them what their programs are for, neither of them give explanations. Matt had previously made it clear that he wanted to use infinity and uses the REPEAT block.

They each run their programs a few times. I ask each one of them about the features they added to KIBO. Matt says he put the gun and wheels on because he "wanted to make

KIBO cool." Jen added the paper with the hearts and other designs to decorate KIBO. Noah explains that the antennas were "so that it can feel when it's touching, like it can feel the floor, or it can feel our hand if we touch it." They run the last program one more time and head out for the day.

Group A Free Play with big-KIBO & KIBO

Three weeks later in their last session with big-KIBO the children were instructed to make whatever they want. I tell them that KIBO will be available half way through the session. They each begin to build their own features to be added to big-KIBO. Noah begins by building a "soot" (Figure 20) which he explains "helps KIBO tow things." As he works on his "soot" it takes on additional functions. He tells me, "I'm going to make it into a tow race car, a race car that tows things." While he continues to work on his design he decides to add antennas to "help KIBO climb." Later his teacher asks what he is working on.

Teacher: do you want to tell me what that thing is?
Noah: It's a spoiler that helps tow things
Teacher: Oh it tows things
Matt: Spoilers usually makes cars don't spin
Teacher: Why does it need a spoiler
Noah: So that KIBO can go fast

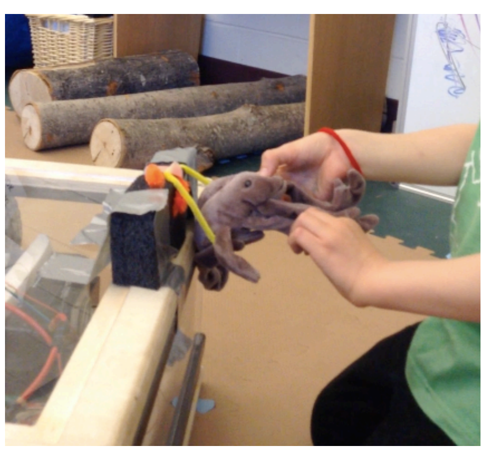

Figure 20. Crawl In the "Soot"

Once he is satisfied with his creation Noah begins to program. He builds the program BEGIN, SHAKE, SHAKE, FORWARD, FORWARD, END (Figure 21). I ask him about his work.

> Me: What does it do?
> Noah: This one is rock climbing [pointing at the first SHAKE]. This one is rock climbing [pointing at the second SHAKE]. This one is racing [pointing at the first FORWARD]. This one is racing [pointing at the second FORWARD].
> Noah: But I need more racing [He gets up and looks for more Forwards blocks].

Figure 21. Noah Presents his Climbing and Racing Program

The group spends the rest of the session making a variety of features for big-KIBO. Jen makes which can carry small objects. Matt makes a way finder, LEGO pieces and a ping pong ball that have been wrapped in a lot of duct tape, that helps KIBO feel things and find its way. Matt also made a “singbat” which he also calls a “musical rat”, a Rigamajig disk with many wing nuts in it that is a “musical thing.” In total there were 10 distinct features added to big-KIBO (Figure 22) in this paly session. Including a KIBO.

Figure 22. Group A Final Project With big-KIBO

When it came time to program their final creation Matt made a program that included REPEAT but had no other explicit meaning. When they ran the program Noah followed big-KIBO pulling on the string attached to his "wingbat" (Figure 23).

Figure 23. Noah Operates The "Wingbat" While big-KIBO Runs a Program

Play Findings

Developmentally kindergartners are still mastering the ability to "transfer some fantasy in the head to some play medium" (Scarlett et al., 2004) in the world. Both KIBO and big-KIBO offer unique affordance as play mediums. From the sessions described there appear to be differences and similarities in the structure and content of children's play with the KIBO and big-KIBO.

Big-KIBO seemed to facilitate more structurally complex narratives than KIBO. Kiki had a birthday, clothing, and was spoken to as if it were actually a kitty. When the same group turned KIBO into a cat it merely had ears. When Noah programs big-KIBO to climb and race he is giving symbolic meaning to actions. Noah's symbolic objects ("soot" and antenna) now have accompanying symbolic actions, this is not present in any other part of the data set. While one could argue that this is a function of his developing programming skills, that would only serve to further support this structural difference.

Another seemingly structural difference in the way children play with the robots is in the use of their bodies. Big-KIBO invites—and sometimes requires—children to engage with their whole bodies. This is evident even from the first time they meet the robots. Those who were introduced to big-KIBO first rested their bodies on its frame while investigating its features (Figure 24). Those who met KIBO first sat back and pointed with their fingers.

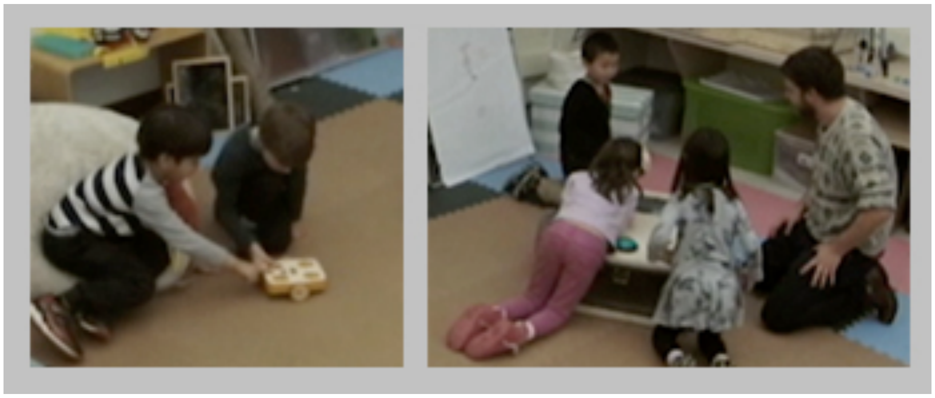

Figure 24. Meeting KIBO and big-KIBO For the First Time

Its size allows children to relate to it in a different way, at times they hugged or kissed big-KIBO—KIBO got no such love. When big-KIBO is active it requires everyone in the rooms attention—at the very least for their safety. The ability to ride big-KIBO opens up new opportunities for the types of things it can do, although this might be more of a content difference than a structural one. No strong examples of the action of riding big-KIBO having symbolic meaning emerged from the data.

A content difference that was observed was in the ways that children engaged their bodies in the testing of the robots' physical limitations. KIBO can be picked up (even dropped) flipped over, and ran on its side. Big-KIBO requires all of a child's strength or coordinated teamwork to move (Figure 25). They may also discover that big-KIBO has the power to move bookshelves, and that KIBO's gears start to grind when it runs into a wall.

Figure 25. Hadar Moving Each Robot

The content of children's creations were sorted into four categories that emerged as themes from the data: functional features, features that are representations of known objects, features that are novel imaginative creations, and decorations. Jen's trunk, and Vinh's handlebars are functional features that appeared in play with big-KIBO. There are no examples of functional features with KIBO. Matt's way finder and Kiki's eyes, ears, and tails all are all representations of known objects that were built for big-KIBO. Matt's gun and the cat ears are examples found for KIBO. Both the "singbat" and the "wingbat" are examples of novel imaginative creations that were made for big-KIBO. There are no examples of this type of creation found in the data for KIBO. Liana made decorations for both KIBO and big-KIBO they took the form of drawings on construction paper that were then taped on to the robots.

We know good play when we see it (Scarlett et al., 2004). The examples in this section show children having fun while hard at work playing with what is meaningful to them. The children themselves make this clear by asking to keep their robots how they are at the end of their free play sessions.

Chapter Seven: Discussion

As an exploratory study this project has accomplished its goal of generating many questions. In this section I discuss my initial findings, while the large set of questions and future directions are addressed in the next chapter.

This thesis has shown that big robots can provide *little* kids new opportunities to engage in computational thinking. While there were no explicit differences in the stages of programming or types of computational thinking that occurred between big-KIBO and KIBO, it is also clear that big-KIBO is not merely a bigger canvas for children to cast their imaginations on. Big-KIBO provides children with new play and coding experiences. Big-KIBO provides children with an opportunity to add features that have functions in the real world, where KIBO is otherwise too small to accommodate (e.g. Noah's "soot" for towing). Big-KIBO engages children in gross motor play while programming. The ability to ride their own creations is a feature previously not available for children. Big-KIBO also appeared to facilitate more complex narratives within children's fantasy play.

Big-KIBO is not meant to be better than KIBO. It is meant to offer new opportunities for children to engage with a wider range of digitally mediated materials. Much like the wide range of non-digital materials we know are important for their development (Curtis & Carter, 2014).

The ideas of constructionism are very present in the children's work with big-KIBO. "Kiki" and the "soot" are examples of children's hard work towards creating personally meaningful projects while developing their programming skills, computational thinking, and crafting. Big-KIBO supports both the bricoleur and the top-down maker

(Turkle & Papert, 1990; Papert & Harel, 1991;Worsley & Blikstein, 2013; Resnick, 2006; Vizner & Strawhacker, 2016). For the bricoleur it provides an even bigger stage for ideas to come together (e.g. Group A's play with each robot). For the top-down maker it offers more space to add more features to liken their creation to their goal (e.g. Kiki) and/or to create larger representations than previously available (e.g. Lucas's chinchilla head). While I only engaged with the children for one session a week, their thinking about the robots persisted. Anecdotally I met Noah's mother who told me that Noah had been working to build a map so that the robot could find its way to their house.

The Maze activity—although initially added to this project in order to investigate perspective taking—showed how Bers' (2018) powerful ideas of computational thinking are important to the children's work with both KIBO and big-KIBO. Combining these powerful ideas with a developmental model of programming provides a strong framework for investigating the types of learning that happen with both KIBO and big-KIBO. Given the scale of this study it was difficult to generate many examples of each of the developmental stages provided, however, a clear picture of what it means to program at the programming stage has been presented.

This project begins to fill the gap in academic research by showing how scale affects the way young children play with robots. It also extends the literature by bringing a developmental approach to the importance of computational thinking in early childhood.

Chapter Eight: Limitations & Future Work

Given the time frame and resources available this study was limited in the scope to which it was able to address the question of "how are children's experiences with big-KIBO different from KIBO?" Together with what was accomplished these limitations have given shape to many new questions, ideas, and exciting directions. The limitations and future work are divided into three parts, big-KIBO itself, the study, and the models that were generated in the analysis.

Big-KIBO

Big-KIBO was built by one graduate student in one semester. This production timeline required me to be selective of the features that were most important to copy from KIBO. I was not able to reproduce the feature that allows KIBO to rerun a stored program. This inevitably changed the way children could execute as well as debug programs. While it was unintentional, the contrast to the repeated pressing of the run button on the KIBO served as an important reflection point when developing the proto-programmer stage. It also provided for Hadar's discovery of the END hack. Further it draws our attention to the importance of the control paradigms we design for children.

KIBO has a set of sensors (distance, sounds, and light) that were not used in this study. They support more complex computational thinking, they add additional control structures, and provide more opportunities for experimentation. It would be exciting to see what children would do if big-KIBO had sensors. Would the mastery of a distance sensor have allowed Hadar, Liana, and Vinh to create a program to "catch fish?" Perhaps Noah could have incorporated a wait for clap to transition his racecar from climbing to racing. Sensors would provide a wider set of opportunities for young makers to express more complex ideas. Sensors would certainly have implications for the developmental model (Table 2) and the actions of computational thinking (Table 3) presented in this thesis; it would server to make the models more applicable outside of the KIBO platform.

Big-KIBO represents a new direction of what is possible in designing for young children. They have big ideas, and they can make meaningful projects from large objects. They also seem to enjoy it. Once Matt and Noah greeted big-KIBO by running across the ECMS screaming, “we missed you big-KIBO” and jumping on it for a hug (Figure 26). Once creating big-KIBO would have taken much longer, more engineers, and a more significant budget. Given the state of rapid prototyping and the rise of the maker movement (Dougherty, 2013), I was able to build big-KIBO with the help of a few friends, in one semester, for about 500 USD. This opens the field to the question of what else can be made, at what scales, to support young children’s learning?

Figure 26. Group A Great big-KIBO With Hugs

The Study

Population & Scheduling

The children recruited from this study all came from the Kindergarten classroom at EPCS. While EPCS does its best to admit a diverse group of students their population is a representation of the local area. EPCS students receive some of the best early childhood instruction in the country. Repeating this study with children from other schools may uncover new ways of using big-KIBO.

Schedule and time of day are important to a six year old, even if they can't tell you why. The Kindergartners in this study came to the ECMS to participate in this study from 12:30-1:30. In their regular daily schedule this time came after lunch and replaced rest. Not an ideal time. The children were either very tired or very restless. Much of the teacher's and my energy was spent helping them regulate their behavior in general. This certainly impacted the amount of time they were able to spend on challenging tasks, although there is no way to measure this. Future studies should take this into consideration.

Research Design

One of the goals of this study was to describe the way children's programming with the KIBO robotics platform develops. In Table 4 there is a clustering of examples in the programming stage. This can be explained by a number of factors. The children in this study are at an age where the ability to symbolize is taking root (Richardson, 1998), and so they quickly move past the proto-programming stage and have the capacity to program with meaning in the programming stage. While children learned and improved in their ability to solve problems and express themselves I imagine there simply was not enough time for them to reach the fluent-programming stage. Additionally the rigid structure of four out of the six activities may have prevented the children from necessary exploration. A study—or studies—with more participants, of varying ages, and over different time scales would generate a stronger more generalizable model.

While the study contained two free play activities, and children brought playfulness to much of their work, at times the research setting stifled play. In reviewing the data I noticed that as a facilitator I was too concerned with getting the children to program that at times I pushed them away from how they wanted to play. For instance when Hadar and Liana wanted to program big-KIBO to hunt for fish and I convinced them to program it to crawl instead. To this end it would be interesting to explore how children use big-KIBO in other settings (e.g. classroom, home, museums). Play is inherently hard to scope and describe. Bringing KIBO into new places to play would help to create sharper tools for description and assessment.

Structure of Activities

The activities, and their order, were selected to facilitate the children's getting know the robot. The first task, Robot Dance Party, has elements of both extrinsically determined specific tasks—programming the hokey pokey—and also moments for intrinsically motivated free expression with the robot—make your own dance move. In this session children are exposed to the robot as an object that can be programmed to do something that someone else asks them to do, or something they want it to do. In a sense easing them in to the full range of what's possible.

The second task, Robot Mazes, is rigid and highly structured. Requiring the children to solve explicit tasks that have right and wrong answers. Presumably this "teaches" them how to control the robot to do a specific, in this case externally dictated, task. This task is designed for the children to get a handle on how to get the robot to do something very specific.

When they reach the third session, Free Play, they presumably know how to program KIBO to do an explicit task whether it be internally or externally dictated. With this "knowledge" it is their time to cast their imaginations onto the robot and use it to play, i.e. explore topics that are personally meaningful to them with KIBO.

Moving forward the effects that the activities themselves have on the way children develop, learn, and play with KIBO should be examined. A study in which the activities are presented in different orders would provide insight into how to best scaffold activities for young children learning to use new technologies. A study in which each group is presented with only one type of activity would help to understand the importance of each type of activity.

Data Analysis

This study generated a large pool of data, given the scope of this thesis I am certain that there is much more to be discovered with in it. Future researchers could take a quantitative approach using software such as BORIS (Friard & Gamba, 2016) to keep track of frequencies of specific behaviors related to the development of programming, computational thinking, and play narratives. Such software may be used to generate transition flow diagrams to better understand the relationships between stages of development and behaviors of computational thinking. It may also shed light on the role of the facilitator in each of the sessions.

This thesis presents snapshots of sessions involving small groups of children; it does not present a complete developmental trajectory for any one child. A case study presenting a detailed narrative of one student from each group would support, or perhaps refute, the conclusions reached in this thesis.

Developmental Model of Computational Thinking

The work done in this thesis to organize the development of programming a KIBO is a model for what might be done with other programming languages and robotics platforms. The acts required to program will be different (e.g. pressing buttons on Bee-Bot, or dragging blocks on a tablet for Scratch Jr) but the trajectory of learning to program with them should be the same because programming languages and robotics platforms are mediums of expression. What develops in my developmental model is the ability to express ones self—to program—with KIBO.

Instead of creating a model for each programming language and robotics platform it would be more powerful to create a developmental model of computational thinking. It is computational *thinking*—with the emphasis on the *thinking*—that is expressed with a given computational medium; a developmental model of computational thinking would be applicable to all programming languages and robotics platforms. Describing the acts of each of the seven powerful ideas of computational thinking from Bers (2018) as they develop would be a strong starting point for this endeavor.

This work would require observing the development of computational thinking in a number of mediums. One could begin this work by using the structure of Table 4 and replacing the examples I have provided with the specific acts of each of the powerful ideas at each developmental stage and repeating this task with several computational mediums. This type of work would have wide implications, in education, policy, assessment, and the development of new technologies.

Chapter Nine: Conclusion

I am the first to admit that big-KIBO is an absurd object, that's what makes it exciting and important. Absurdity, the juxtaposition of the expected with the unexpected, is exciting. With big-KIBO I play this game with scale. It is what made *Honey I Shrunk the Kids* so exciting. I have always found the physical manifestation of the absurd thought provoking. To put it in Piagetian terms absurdity forces one into a state of disequilibrium. It pushes one to reimagine what can and can't be.

I have described big-KIBO to hundreds of people around the world; their responses are often a mixture of delight and disbelief. Modern perceptions of robots are humanoid (e.g. Wall-E, Bender, and the robots of Westworld), militaristic (e.g. DARPA and Terminator), or domestic (Roomba and Rosie from *The Jetsons*). They are not often thought of as something to be riden or used by children. Big-KIBO is a playful violation of this norm. It is an invitation to wrestle with what it means to be a robot, especially one for children to learn with.

The goal of this project is to give children an object that allows them to reimagine what can and can't be. As computing becomes ubiquitous in our society and embedded in our every day devices (e.g. smart devices, internet of things, Siri and Alexa, etc.), we must prepare our children to think beyond programming computers in the traditional sense and begin to see the world as programmable. The power to design, make, and control objects in ones own world comes from a strong foundation in computational thinking. As consumerism rises children and adults must be reminded that things are made by people, and that they too have the power to make and control them.

It was a very powerful moment for Matt when he discovered that I built big-KIBO, he looked at me in total disbelief. I hope that in that moment he realized that he too has the power to imagine and create that which is meaningful to him. Technology has the capacity to pacify or empower, to be a playpen or a playground. This project is a FORWARD step in understanding the types of technologies that can best serve young children.

References

Ackermann, E. (2003). Hidden drivers of pedagogic transactions: Teachers as clinicians and designers. In *Proceedings of Eurologo* (pp. 29–37). Retrieved from http://web.media.mit.edu/~edith/publications/2003-Hidden.drivers.pdf

American Academy of Pediatrics Announces New Recommendations for Children's Media Use. (n.d.). Retrieved March 8, 2017, from 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

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.

Bender, J. (1978). Large hollow blocks: Relationship of quantity to block building behaviors. *Young Children*, *33*(6), 17–23.

Bers, M. U. (2008). *Blocks to Robots: Learning with Technology in the Early Childhood Classroom*. New York: Teachers College Press.

Bers, M. U. (2012). *Designing Digital Experiences for Positive Youth Development: From Playpen to Playground* (First Edition). Oxford, New York: Oxford University Press.

Bers, M. U. (2018). *Coding as a Playground: Programming and Computational Thinking in the Early Childhood Classroom*. Routledge. Retrieved from https://www.routledge.com/Coding-as-a-Playground-Programming-and-Computational-Thinking-in-the-Early/Umaschi-Bers/p/book/9781138225626

Bers, M. U., & Horn, M. S. (2010). Tangible Programming in Early Childhood. *High-Tech Tots: Childhood in a Digital World*, *49*, 49–70.

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.

Blikstein, P. (2015). Computationally Enhanced Toolkits for Children: Historical Review and a Framework for Future Design. *Foundations and Trends® in Human–Computer Interaction*, *9*(1), 1–68. https://doi.org/10.1561/1100000057

Campbell, S. F. (Ed.). (1976). *Piaget Sampler: An Introduction to Jean Piaget Through His Own Words*. New York: John Wiley & Sons Inc.

Curtis, D., & Carter, M. (2014). *Designs for Living and Learning, Second Edition: Transforming Early Childhood Environments* (2 edition). St. Paul, MN: Redleaf Press.

Dougherty, D. (2013). The maker mindset. In M. Honey & D. E. Kanter (Eds.), *Design, make, play: Growing the next generation of STEM innovators* (pp. 7–11). Routledge. Retrieved from https://llk.media.mit.edu/courses/readings/maker-mindset.pdf

Early Childhood Robotics Network. (n.d.). Retrieved July 6, 2017, from http://tkroboticsnetwork.ning.com/

Edwards, C. (2002). Three Approaches from Europe: Waldorf, Montessori, and Reggio Emilia. *Faculty Publications, Department of Child, Youth, and Family Studies*. Retrieved from http://digitalcommons.unl.edu/famconfacpub/2

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. https://doi.org/10.1080/07380569.2016.1216251

Elkind, D. (2003). Montessori and constructivism. *Montessori Life; New York*, *15*(1), 26.

Feldman, D. H. (1994). *Beyond Universals in Cognitive Development, 2nd Edition* (2 edition). Norwood, N.J: Praeger.

Feldman, D. H. (2004). Piaget's stages: the unfinished symphony of cognitive development. *New Ideas in Psychology*, *22*(3), 175–231. https://doi.org/10.1016/j.newideapsych.2004.11.005

Friard, O., & Gamba, M. (2016). BORIS: a free, versatile open-source event-logging software for video/audio coding and live observations. *Methods in Ecology and Evolution*, *7*(11), 1325–1330. https://doi.org/10.1111/2041-210X.12584

Hewett, V. M. (2001). Examining the Reggio Emilia Approach to Early Childhood Education. *Early Childhood Education Journal*, *29*(2), 95–100. https://doi.org/10.1023/A:1012520828095

In Memory: Seymour Papert | MIT Media Lab. (n.d.). Retrieved March 6, 2017, from https://www.media.mit.edu/people/in-memory/papert

Kaplan, B. (1983). Genetic Dramatism: Old Wine in New Bottles. In S. Wapner & B. Kaplan (Eds.), *Toward a Holistic Developmental Psychology* (pp. 53–74). Hillsdale, NJ: Lawrence Erlbaum Associates.

Kazakoff, E. R., 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. https://doi.org/10.1007/s10643-012-0554-5

KIBO. (2014, September 12). Retrieved March 8, 2017, from http://kinderlabrobotics.com/kibo/

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

McDonald, L., & Stuart-Hamilton, I. (2003). Egocentrism in Older Adults: Piaget's Three Mountains Task Revisited. *Educational Gerontology*, *29*(5), 417.

Paley, V. (2004). *A Child's Work: The Importance of Fantasy Play*. Chicago, IL: The University of Chicago Press.

Papert, S. A. (1993). *Mindstorms: Children, Computers, And Powerful Ideas* (2 edition). New York: Basic Books.

Papert, S., & Harel, I. (1991). Situating Constructionism. In *Constructionism*. Ablex Publishing Corp. Retrieved from http://www.papert.org/articles/SituatingConstructionism.html

Perlman, R. (1976). Using computer technology to provide a creative learning environment for preschool children. MIT Artificial Intelligence Laboratory Publication 260.

Piaget, J. (1961). The genetic approach to the psychology of thought. *Journal of Educational Psychology*, *52*(6), 275–281. https://doi.org/http://dx.doi.org/10.1037/h0042963

Resnick, M. (2006). Computer as Paintbrush: Technology, Play, and the Creative Society. In D. Singer, R. Golikoff, & K. Hirsh-Pasek (Eds.) (pp. 192–206).

Oxford University Press. Retrieved from http://www.audentia-gestion.fr/MIT/playlearn-handout.pdf

Richardson, K. (1998). *Models Of Cognitive Development* (1 edition). Hove, East Sussex, UK: Psychology Press.

Rufo, D. (2012). Building forts and drawing on walls: Fostering student-initiated creativity inside and outside the elementary classroom. *Art Education*, *65*(3), 40–47.

Scarlett, W. G., Naudeau, S. C., Salonius-Pasternak, D., & Ponte, I. C. (2004). *Children's Play* (1 edition). Thousand Oaks: SAGE Publications, Inc.

Scarlett, W. G., & Wolf, D. (1979). When It's Only Make-Believe.pdf. *New Directions for Child Development*, *6*.

Strawhacker, A., Sullivan, A., & Bers, M. U. (2013). TUI, GUI, HUI: is a bimodal interface truly worth the sum of its parts? In *Proceedings of the 12th International Conference on Interaction Design and Children* (pp. 309–312). ACM. Retrieved from http://dl.acm.org/citation.cfm?id=2485825

Sullivan, A. A., Bers, M. U., & Mihm, C. (2017). Imagining, Playing, and Coding with KIBO: Using Robotics to Foster Computational Thinking in Young Children. *Proceedings of of the International Conference on Computational Thinking Education*. Retrieved from http://ase.tufts.edu/devtech/publications/Sullivan_Bers_Mihm_KIBOHongKong%20.pdf

Sullivan, A., & Bers, M. U. (2017). Dancing robots: integrating art, music, and robotics in Singapore's early childhood centers. *International Journal of Technology and Design Education*. https://doi.org/10.1007/s10798-017-9397-0

Sullivan, A., Elkin, M., & Bers, M. U. (2015). KIBO robot demo: engaging young children in programming and engineering (pp. 418–421). ACM Press. https://doi.org/10.1145/2771839.2771868

Thelen, E., & Smith, L. B. (1998). Dynamic systems theories. *Handbook of Child Psychology*. Retrieved from http://onlinelibrary.wiley.com/doi/10.1002/9780470147658.chpsy0106/full

Turkle, S. (1984). *The Second Self: Computers and the Human Spirit* (1st edition). New York: Simon & Schuster.

Turkle, S., & Papert, S. (1990). Epistemological pluralism: Styles and voices within the computer culture. *Signs: Journal of Women in Culture and Society*, *16*(1), 128–157.

Vizner, M., & Strawhacker, A. (2016). Curious Construction Kit: A Programmable Building Kit for Early Childhood. In *Proceedings of the 6th Annual Conference on Creativity and Fabrication in Education* (pp. 90–93). New York, NY, USA: ACM. https://doi.org/10.1145/3003397.3003412

Vygotsky, L. S. (1978). *Mind in Society: The Development of Higher Psychological Processes*. (M. Cole, V. John-Steiner, S. Scribner, & E. Souberman, Eds.) (Revised ed. edition). Cambridge, Mass. u.a.: Harvard University Press.

Ware, E. A., Uttal, D. H., & DeLoache, J. S. (2010). Everyday scale errors. *Developmental Science*, *13*(1), 28–36. https://doi.org/10.1111/j.1467-7687.2009.00853.x

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

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *366*(1881), 3717–3725. https://doi.org/10.1098/rsta.2008.0118

Worsley, M., & Blikstein, P. (2013). Programming pathways: A technique for analyzing novice programmers' learning trajectories. In *International Conference on Artificial Intelligence in Education* (pp. 844–847). Springer. Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-39112-5_127

Appendix A: Group B Solves Mazes With big-KIBO

In their second session with big-KIBO Vinh, Liana and Hadar were asked to solve the three mazes made out of floor tiles. They were particularly rambunctious during this session. At times the teacher separated the rest of the group from Hadar so that she could focus on the work. She did the majority of the programming. Here I present excerpts from their session.

Maze 1

Me: What do you think it would take to get KIBO from here at the beginning to here at the end [pointing to the beginning and end of the maze]
Hadar: I do
Me: Do you want to do it? Can you get KIBO to move from where he is to the end of the blue
Hadar: [Tries to drag big-KIBO]
Teacher: No Program
Hadar: [Holds blocks to her face]
Me: Last week I saw you programming by putting the blocks together, this week I don't think it works if you put them against your face.
Hadar: [Builds the program BEGING, FORWARD, FORWARD, END]
Me: What do you think is going to happen if you do this [pointing to program]? Will it help KIBO get from here to the end?
Hadar: Ya
Me: Ya, what's he going to do?
Hadar: [scans half the program and gives Liana to scan the other half]
Me: Do you think KIBO is going to make it from here to the end of the path?
Hadar & Liana: Ya
Hadar & Liana: [get on big-KIBO and run the program. big-KIBO goes half way (it looks like it only did one FORWARD, maybe they made a scanning error)]
Me: Did he do your program?
Hadar: [laughing] yes
Teacher: did it go all the way to the end
Hadar: [laughing] yes
Liana: [laughing] no
Teacher: No, Liana says no
Me: Is that the end?
Hadar: Yes
Me: So what do you think you need to add to make it go to the end
Liana: More!
Me: More forwards?

Hadar: More! [runs to bucket of blocks]
Me: [Speaking to Liana] can you put it back at the start
Liana: [Tries to move robot]
Hadar: [comes back with more blocks she adds to her program]
Me: How many forwards do you think its going to take
Hadar: This many
Me: That many, do you want to try it?
Me: why did you add the BACKWARDS to the program?
Hadar & Liana: [Laugh]
Hadar: [Removes the BACKWARDS]
Teacher: Do you need that?
Liana: [Laughing] yes, we need a backwards

Hadar brings over more FORWARD blocks from the bin and makes the program BEGIN, FORWARD, FORWARD, FORWARD, FORWARD, FORWARD, END. Liana gets on and Hadar scans. They run the program and big-KIBO passes its goal, runs off the path and into the beanbags.

Me: I think it went too far.
Liana: But we made the line
Me: Can you pull him back for me
Liana: We need the BACKWARD
Vinh: [Joins and they push big-KIBO back to the start]
Me: How many FORWARDS was that one?
Hadar: Five
Me: How many should we try this time
Hadar: A hundred
Me: But we want it to go less distance
Hadar: Three
Teacher: Three she says three
Hadar: No four
Teacher: The two of you plan together. You have to come to an agreement and work together.

They take turns scanning blocks. Hadar and Liana get on big-KIBO. They run the program while Vinh is trying to figure out where to sit. Big-KIBO runs into the pillows again. They push big-KIBO back to the start. This time Hadar makes a program with only two FORWARD blocks. They run the program and big-KIBO stops at the end of the maze.

Maze 2

After being redirected back to the programming task Hadar assembles the program BEGIN, FORWARD, RIGHT, FORWARD, END. Vinh is sitting on top of big-KIBO. Hadar scans each of the blocks but not the END block. Excited she hits the run button. The teacher calms her down and reminds her to scan the END. She scans the END and hits the run button. Big-KIBO turned right to early and came off the path. If it had been positioned a little further forward at the start it would have successfully run the maze.

Teacher: Did it do it?
Vinh: Nooo
Teacher: I think it did, Lets do it again. Does it store more than once or just once?
Me: We have to scan it again. Vinh can you help me put it back to the beginning?
Vinh: Ok but this is a little heavy I need help.
Me: I'll help you I know you are really strong. Together I think we can do it
Teacher: I'll hold your paper and octopus

Vinh walks away to put his papers down while the teacher and I reposition big-KIBO at the start of the maze. Meanwhile Hadar has reassembled her program. She brings it over to big-KIBO and scans it, throwing the blocks to the side as she goes. She hits the run button then hops on. Big-KIBO successfully reaches the end of the maze. Hadar jumps off and says "yay now we get to do infinity!" in anticipation of learning to use the REPEAT block.

Maze 3

I ask Hadar to show me her solution to Maze 2. She moves around the room to find the pieces. Then I ask:

Me: Does this shape [pointing at Maze 2] look like part of that shape [pointing at maze 3]
Hadar: [Nods in agreement]
Me: What's different
Hadar: It has two of these
Me: It has two of these?
LM: Ya, two of these right
Hadar: [Nods in agreement]
Me: And this one [pointing at maze 3] is a little bit longer than this one [pointing at maze 2] right?

Hadar looks up at maze 3 then assembles the code chunk FORWADR, RIGHT, FORWARD, FORWARD. She passes it to her teacher and says you do it. She sits down in a beanbag chair and has a small tantrum saying she is tired and that the task is too hard. The teacher calms her down offering to do the work together. The teacher sends Vinh and Liana away who are playing on big-KIBO and distracting Hadar. Once Hadar has calmed down the Teacher introduces the REPEAT block.

Teacher: When it says repeat [places REPEAT block at the front of Hadar's code chunk] oh I am supposed to repeat something and then when you put it at the end [places END REPEAT block at the end of the code chunk]. So KIBO will say oh when it says this [points at REPEAT] I'm going to get a set of commands or instructions [points at Hadar's code chunk] but I don't know how many times. How many times? [Hands Hadar the INFINITY parameter].
Hadar: [Places the INIFNITY parameter on the REPEAT block].
Teacher: Forever. KIBO is going to say I'm going to repeat this. I'll do This this this this, and then I'm going to do this this this this, and then I'll go this this this this, I'll do this this this this, I'll do this this this this forever, I'm going to this this this forever, Im going to this this this forever, and that's how it reads. [She points to each of the blocks in Hadar's code chunk sequentially each time she says this].
Teacher: Is that what you want? So you only need two more pieces to make it work [sets down a BEGIN and END block in front of her].
Hadar: [attaches the blocks then goes to get her friends]
Teacher: There you go we worked on it together. We need to scan now, we need to scan now Hadar.
Me: [to teacher] also its going to go off, it has too many FORWARDs
Teacher: [to Hadar] Miki thinks that we have to take one FORWARD out.
Me: Hadar can I show you something with your body.
Me: Do you remember here how many FORWARDS it took [walking across maze 1] one, two.

Me: And over here we have the same number right [walk along one edge of maze 3] one, two. But how many forwards do you have in your program total?
Hadar: one two three four [pointing to blocks as she counts]
Teacher: But how many FORWARDs. How many go FORWARD just straight line FORWARD.
Hadar: One, two, three
Me: So we need to take one out right.
Hadar: Takes out one FORWARD making the sequence in the REPEAT FORWARD RIGHT FORWARD.
Me: Can I ask you one more Question do we go FORWARD FORWARD then RIGHT or FORWARD RIGHT FORWARD [I walk in these directions on the maze as I say them out loud].
Vinh: FORWARD FORWARD RIGHT [steps these directions as he says them.
Me: Can you help Hadar look at her program?

Vinh helps Hadar with her program making the final program BEGIN, REPEAT (INFINITY), FORWARD, FORWARD, RIGHT, END REPEAT, END. Hadar scans it. As Vinh and Hadar get on I joke that they will be there forever. Hadar gets nervous but her teacher reassures her that she won't let that happen and that we can turn it off. They hit the run button but nothing happens. There is another class that needs to enter the space so I quickly rescan their program. They run it and travel around the Maze 3 a few times before I turn off big-KIBO.