Journal of Research in STEM Education

Vol 7, No 1, July 2021, 47-68

https://doi.org/10.51355/jstem.2021.102

RESEARCH REPORT

Assessing Robotics Skills in Early Childhood: Development and Testing of a Tool for Evaluating Children's Projects

Madhu Govind[1], Marina Bers

Tufts University, USA

Abstract: *Children's robotics skills can be assessed in various ways, one being examining the unique projects that they create. This paper discusses the multi-phase development and testing of a robotics project rubric. The rubric considers both the programming concepts and the aesthetic design elements of a project, which enables researchers and practitioners to determine the overall level of complexity exhibited in the robotics project. This paper presents the background literature and theoretical framework that contributed to the rubric design and summarizes findings from iteratively developing and testing the rubric with a total of 173 robotics projects. Implications for future research and practice are also discussed.*

Keywords: *Rubric, project-based assessment, programming, robotics, early childhood*

Introduction

Ever since Seymour Papert and colleagues' invention of the LOGO turtle in the late 1960s, there has been an increasing focus on the possibilities for robotics to transform young children's thinking and learning. Although shocking at the time, Papert's constructionist ideas about children actively producing and sharing their own technological artifacts are now widely embraced in today's increasingly digital and global society (Bers, 2020; Papert & Harel, 1991). To date, there are over 34 computational kits targeted for children under the age of seven, many of which are physical robots with tangible programming components (Yu & Roque, 2018). One

[1] Corresponding Author: Madhu Govind, Tufts University, Eliot-Pearson Department of Child Study and Human Development, 105 College Ave. Medford, MA 02155. Email: madhu.govind@tufts.edu

To cite this article: Govind, M. & Bers, M. (2021). Assessing robotics skills in early childhood: Development and testing of a tool for evaluating children's projects. *Journal of Research in STEM Education, 7*(1), 47-68. https://doi.org/10.51355/jstem.2021.102

ISSN: 2149-8504 (online)

important reason for the popularity of tangible robotics kits is the opportunity to extend the long-lasting tradition of hands-on learning with manipulatives in early childhood education (Bers, 2008; O'Malley & Fraser, 2004; Resnick, 2007). Studies have identified benefits to introducing robotics at an early age, such as making children's learning visible, sparking interest in coding, and supporting STEAM (Science, Technology, Engineering, Arts, and Mathematics) integration (Benitti, 2012; Bers et al., 2013; Horn & Bers, 2019). With the growing popularity of robotics for young children and the advancement of developmentally appropriate robotics kits, there is a need to understand what children can learn from using and creating with these technologies.

Children's learning of and through robotics can be assessed using various methods, such as multiple-choice questionnaires, design scenarios, artifact-based interviews, and project analyses (Brennan & Resnick, 2012). There are benefits and limitations to each method. For example, questionnaires allow for standardization and measurement of discrete skills. Design scenarios and artifact-based interviews are subjective measures but allow for more nuanced assessment of children's conceptual understanding. Project analyses offer insight into the conceptual encounters a child may experience over the course of designing their project; however, encounters do not necessarily equate to mastery. Because of the varying strengths and limitations of each approach, many researchers often recommend using a "system of assessments" (Grover, 2017) to provide a more holistic understanding of children's knowledge. However, quantity should be paralleled with quality. Assessments must be developmentally appropriate (in this case, for young children) and demonstrate purposeful value for research and practice.

This paper presents work that fills a gap in assessment approaches for robotics kits for young children, most specifically, KIBO, which is a screen-free robot that young children program using tangible wooden blocks and personalize using arts and crafts materials and that is been used worldwide (Albo-Canals et al., 2018; Bers et al., 2019; Jurado et al., 2020). Studies have shown that children as young as three can play with KIBO to acquire foundational coding, robotics, and computational thinking skills (Elkin et al., 2016; Relkin et al., 2020; Sullivan et al., 2017). Current assessment approaches include multiple-choice oral questionnaires such as KIBO Solve-Its and KIBO Mastery Challenges (Hassenfeld et al., 2020; Sullivan & Bers, 2015); artifact-based interviews (DevTech Research Group, 2019; Portelance & Bers, 2015); and design scenarios such as the TACTIC-KIBO Assessment and the KIBO Coding Stages Assessment (Relkin & Bers, 2019; Bers, 2019). Although general assessment rubrics were devised in the past to accompany KIBO lessons and support educators in assessing final KIBO projects (DevTech Research Group, 2018), these rubrics lacked specific scoring criteria and adequate psychometric properties that are necessary for more widespread use in research and educational settings. Thus, the KIBO Project Rubric was developed and is presented in this paper.

The KIBO Project Rubric is a project-based assessment that can be used to examine the level of complexity of programming concepts and design elements exhibited in a KIBO robotics project. This rubric can be adapted to be used with other robotic kits for children. First, this paper summarizes the background literature and theoretical framework that contributed to the rubric design. Second, the paper details the multi-phase, iterative process used to develop and test the rubric with a total of 173 KIBO projects and multiple raters. Finally, the paper presents findings from the development and testing process and discusses the implications and limitations of this rubric in regard to future research and practice use.

Background

Studies on young children's experiences with robotics reveal a variety of benefits for children's learning and development. For instance, learning robotics at a young age can support meta-cognitive thinking and problem-solving skills, referred to in the field as computational thinking (Barr & Stephenson, 2011; Clements & Gullo, 1984; Kafai & Burke, 2014; Seiter & Foreman, 2013; Wing, 2006). Children as young as four and five have shown increased understanding of algorithms, control structures such as repeat loops, and debugging strategies after participating in introductory robotics and programming activities (Elkin, Sullivan & Bers, 2016; Strawhacker & Bers, 2019; Wohl, Porter & Clinch, 2015). Early exposure to robotics, as well as other programmable technologies, may also shape attitudes and help dismantle gender stereotypes about the computing field (Sullivan, 2019).

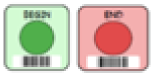
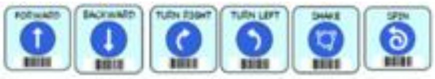
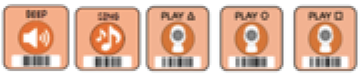
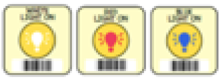
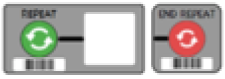
In addition to robotics as a tool for engaging with coding and computational thinking concepts, there are unique benefits of robotics as tangible programming interfaces for early childhood. The term "tangible", first introduced in the mid-1970s by Radia Perlman with her work on the TORTIS Slot Machine for young children, refers to interfaces that use physical objects and surfaces to manipulate and represent digital information. Physical manipulatives such as screen-free robotics kits promote playful, developmentally appropriate practices already prioritized in early childhood education (Brosterman, 1997; Meacham & Atwood-Blaine, 2018). Several studies have found robotics tools more inviting and engaging to young children as compared to unplugged or screen-based tools (Pugnali, Sullivan & Bers, 2017; Strawhacker & Bers, 2015; Wohl, Porter & Clinch, 2015).

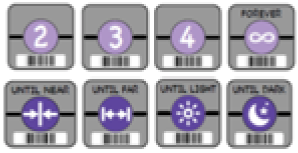
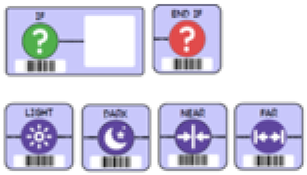
The rubric described in this paper was designed to evaluate projects created with the KIBO robotics kit, a screen-free robotics platform for young children that utilizes tangible programming (Bers, 2020). Children program the KIBO robot by using the barcode scanner embedded in the robot body to scan a series of barcode stickers on tangible wooden blocks. The kit contains sensors, modules, and art platforms so that children can explore more advanced programming concepts, such as repeat loops and conditionals, and use craft materials to decorate their robotic creations (see Figure 1).

The KIBO robot has been used at large in multiple educational settings all over the world. Studies have shown that children of diverse age groups and abilities have exhibited foundational sequencing and creative problem-solving skills while engaging with KIBO (Albo-Canal et al., 2018; Bers, Gonzalez-Gonzalez, & Armas-Torres, 2019; Sullivan, Bers, & Mihm, 2017; Sullivan, Elkin, & Bers, 2015). Educators have since developed and implemented curricula that integrate KIBO with different content areas, including literacy, math, and social sciences (Elkin, Sullivan, & Bers, 2016; Sullivan, Bers, & Mihm, 2017). However, with coding and robotics education still being a relatively new frontier in early childhood education, there is a need for a reliable project rubric to examine the creative computational artifacts that can be produced with the KIBO robotics kit.

Table 1.

Components of the KIBO Robotics Kit

Category	Components	Image
Hardware	KIBO robot with wheels and motors	
	Input/output modules (Distance, Sound, and Light sensors, Lightbulb, Sound Recorder)	
Software	Begin and End blocks	
	Blue Motion blocks	
	Orange Sound blocks	
	Yellow Light blocks	
	Purple Wait for Clap block	
	Gray Repeat blocks + parameters	

Category	Components	Image
		
	Lavender If blocks + parameters	
Art platforms	Expression module	
	Stationary or rotating art platform	

Previous rubrics developed for KIBO (DevTech Research Group, 2018) are curriculum-specific and lack specific scoring criteria and adequate psychometric properties that are necessary for more widespread research and practice use. For example, they utilize a 3-point rating scale, in which the differentiating scale markers are labeled as “completely meets”, “partially meets”, or “does not meet” the stated project requirement. There are between 2-11 different project requirements, depending on the associated lesson topics. Although rating scales can provide some useful information about the level of complexity exhibited in a project, the scoring can often be subjective and requires extensive training for researchers and educators to use effectively with high reliability. In addition to the limitation of subjectivity, these rubrics were designed for existing KIBO curricula that followed a specific lesson structure and format. Thus, its generalizability across other KIBO curricula is unknown, which threatens the rubric’s validity.

Alternatives to rating scales are analytic scales and holistic scales. Analytic scales have specific descriptions for each point on a scale, which may be more difficult to develop and ensure adequate psychometric properties, but they may be better suited to use across varied settings. Holistic scales provide a single overall evaluation; these scales are used often in educational settings (e.g., “Satisfactory, Unsatisfactory, Needs Improvement”) but provide little-to-no specific feedback unless supplemented with space for descriptive notes. This paper explores these various approaches for assessing KIBO robotics projects in different iterations of the KIBO Project Rubric.

Method

The KIBO Project Rubric was developed using a multi-phase, iterative process. During the first phase of development, scoring criteria were selected based on the general KIBO assessment rubrics (DevTech Research Group, 2018) and other project-based assessment rubrics in the field (e.g., Brennan, Haduong, & Veno, 2020; Grover, 2017; Grover, 2020; Salac & Franklin, 2020). By adapting and remixing existing assessments, similar construct labels were kept consistent and unique characteristics of KIBO were identified and developed into additional scoring criteria. This method for rubric development contributed to the initial face validity of the rubric. This first iteration of the KIBO Project Rubric was tested with *N* = 123 second graders' KIBO programs recorded using stickers in individual design journals at the end of a 12-lesson KIBO robotics curriculum. After discussing scoring discrepancies and limitations of this first rubric with the research team, a second iteration of the rubric with more nuanced scoring guidelines was developed. This second version was tested with *N* = 50 KIBO projects documented using videos, pictures, and/or written descriptions. The following sections describe each phase of the development and testing process in greater detail.

Development and Testing Phase 1

The rubric development process began with creating criteria for general and KIBO-specific programming skills. Because KIBO is first and foremost a robotics platform to introduce foundational programming concepts to young children, the first goal was to capture children's conceptual understanding of the KIBO programming blocks. As such, the first iteration of the KIBO Project Rubric consisted of a single holistic 1-5 scale (5 being the highest possible score) guided by five criteria that comprised general programming skills and project characteristics: syntactical accuracy, number of blocks, types of blocks, use of repeat loops/conditionals, and evidence of purposeful programming (see Table 2). Syntactical accuracy referred to whether or not the program would run successfully when executed in the exact manner as written. Number of blocks referred to the total number of KIBO blocks in the program. Types of blocks referred to the variety of KIBO blocks used in the program (apart from the Begin and End blocks), for instance, blue Motion blocks or gray Repeat blocks. The more blocks used with greater variety, the more complex the project. The use of repeat loops and conditionals was the fourth scoring criterion; if used correctly, the project would receive at least a score of 4. The rationale was that the presence of these advanced programming blocks might indicate a child's more advanced understanding of the KIBO robotics kit. The final criterion was evidence of purposeful programming, that is, if there was any written or pictorial evidence from the child's design journal that the blocks were meaningfully related to their final project idea. An example of purposefulness is the use of the Beep block to represent a car honking its horn.

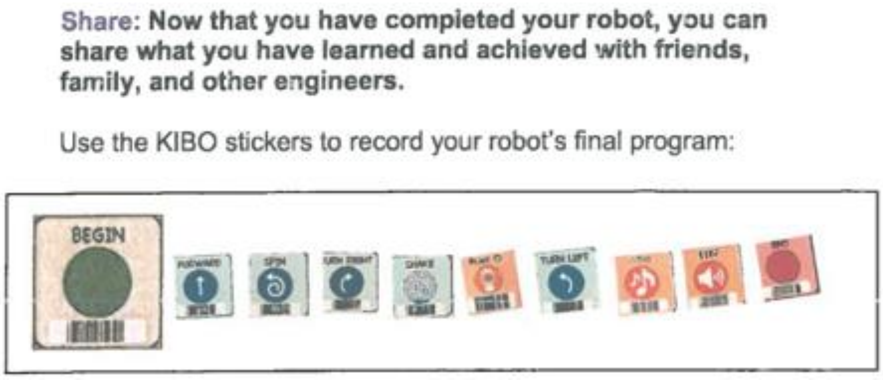
Table 2.

First Iteration of KIBO Project Rubric with Single Holistic Scale

Score	Level of Project Complexity	Syntactically Correct	Number of Blocks	Types of Blocks	Repeats / Conditionals	Purposefulness
1	Budding	No	-	-	-	-
2	Developing	Yes	<=5	1	No	-
3	Proficient	Yes	>5	1-2	No	-
4	Advanced	Yes	>5	2+	Yes	-
5	Distinguished	Yes	>5	2+	Yes	Yes

Before this rubric was tested with KIBO projects, five experts in early childhood technology with extensive KIBO research and training experience reviewed the rubric and provided feedback, which contributed to the construct validity of the rubric. Their feedback led to several changes to the rubric and its format. For instance, the rubric was digitized into an online Google Form so that scores would be automatically populated into a spreadsheet for analysis. Several checkbox-style items were also added in order to capture the source of syntax errors (e.g., incorrect use of Begin or End block, misplaced or missing Begin or End Repeat/If block, misplaced or missing parameter, etc.). The goal was that these added questions would provide a deeper understanding of children’s conceptual encounters and possible misconceptions of key KIBO programming concepts.

This rubric was tested with *N* = 123 KIBO projects recorded in individual design journals with stickers representing the various KIBO programming blocks (see Figure 1). These KIBO projects were completed at the culmination of a 12-hour second grade coding, robotics, and literacy curriculum that was implemented in eight elementary schools in a Virginia public school district. To score these KIBO projects, a research assistant was trained on the rubric. 15% of the projects were jointly scored by two raters to establish agreement and resolve any scoring discrepancies. There was 96% agreement in scores between the two raters.



Share: Now that you have completed your robot, you can share what you have learned and achieved with friends, family, and other engineers.

Use the KIBO stickers to record your robot's final program:

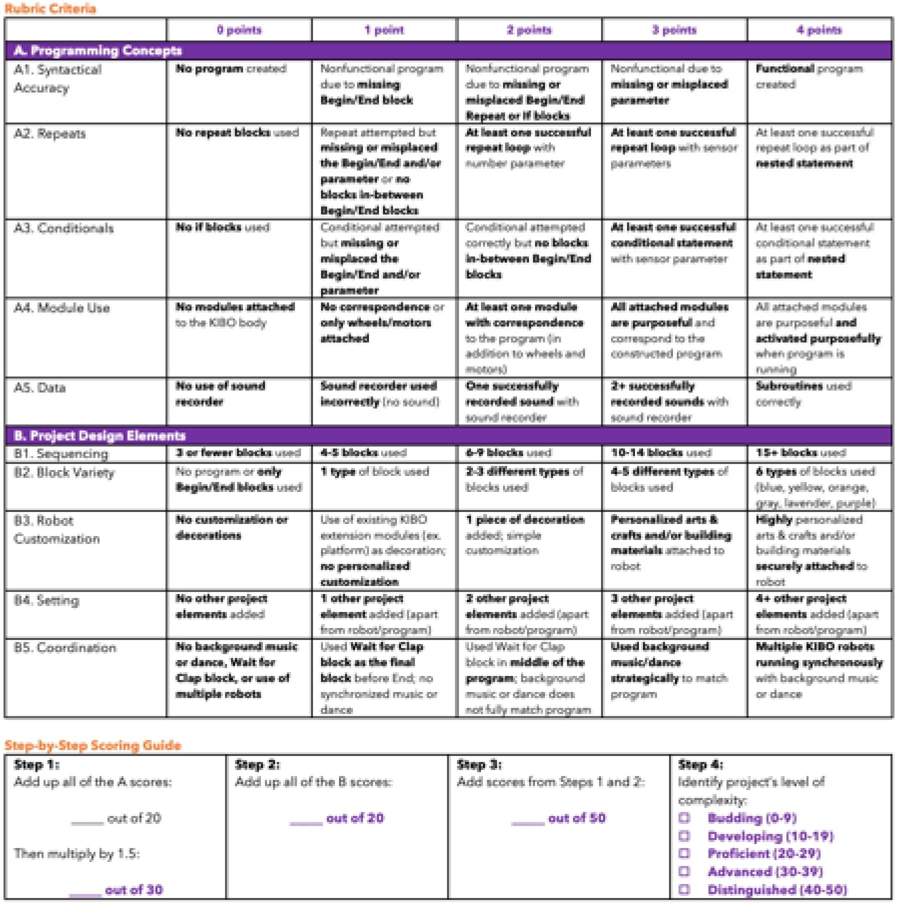
Figure 1. KIBO Programs Recorded Using Stickers in Students’ Design Journals

Percentage agreement was considered to be an appropriate method for assessing reliability at this point in the development process, as several rubric limitations had already been revealed, requiring the need for further revision and retesting. One limitation was the inability to capture the full breadth of KIBO robotics projects from sticker programs alone. For instance, the modules and sensors attached to the robot were unable to be examined, as well as the ways in which the robot was decorated or personalized. These robot characteristics were difficult to assess from design journals but represent a critical component of KIBO robotics projects. Further testing was indeed necessary and required looking at projects that were documented with videos and pictures. Another limitation was that projects were penalized for syntactical inaccuracy, even if advanced programming concepts were attempted. In order to remedy this limitation, the single holistic scale was expanded into an analytic scale with descriptors for each criterion. These individual scores could then be consolidated into a holistic score that would represent the overall level of complexity exhibited in the robotics project. This next section describes this revised rubric development and testing procedure.

Development and Testing Phase 2

The second version of the KIBO Project Rubric consisted of two sets of scoring criteria: (A) Programming Concepts and (B) Project Design Elements. Programming Concepts refer to foundational skills and concepts that are specific to the activity of programming. The five sub-categories of Programming Concepts are (A1) syntactical accuracy, (A2) repeats, (A3) conditionals, (A4) module use, and (A5) data. Project Design Elements refer to project characteristics that add aesthetic appeal, display originality and creativity, or extend the complexity of the project. The five sub-categories of Project Design Elements are (B1) sequencing, (B2) block variety, (B3) robot customization, (B4) setting, and (B5) coordination.

Figure 2 displays the rubric with descriptions of these ten criteria, each of which were scored on a 0-4 scale with individual descriptors for each score on the scale. The higher the points for a particular construct, the more advanced skill exhibited by the project creator. The maximum score for each set of scoring criteria was 20 points. However, when computing a final score for the project, the scores for Programming Concepts and Project Design Elements were weighted differently. Although KIBO offers ample integration opportunities and has aesthetic appeal, its primary educational purpose is to introduce foundational programming concepts to young children. Thus, this rubric utilized a 60-40 weighted ratio, with emphasis given to Programming Concepts by multiplying its summed score by 1.5. Therefore, the maximum number of points for Programming Concepts was 30 points, which brought the total summed score to a maximum of 50 points. Because there was no existing evidence that could be used to determine specific cutoff points for different levels of project complexity, this total score was then categorized evenly into five levels: Budding (0-9 points), Developing (10-19 points), Proficient (20-29 points), Advanced (30-39 points), and Distinguished (40-50 points).



Rubric Criteria

	0 points	1 point	2 points	3 points	4 points
A. Programming Concepts					
A1. Syntactical Accuracy	**No program** created	Nonfunctional program due to **missing Begin/End block**	Nonfunctional program due to **missing or misplaced Begin/End Repeat or If blocks**	Nonfunctional due to **missing or misplaced parameter**	**Functional** program created
A2. Repeats	**No repeat blocks** used	Repeat attempted but **missing or misplaced the Begin/End and/or parameter** or **no blocks in-between Begin/End blocks**	**At least one successful repeat loop** with number parameter	**At least one successful repeat loop** with sensor parameters	At least one successful repeat loop as part of **nested statement**
A3. Conditionals	**No if blocks** used	Conditional attempted but **missing or misplaced the Begin/End and/or parameter**	Conditional attempted correctly but **no blocks in-between Begin/End blocks**	**At least one successful conditional statement** with sensor parameter	At least one successful conditional statement as part of **nested statement**
A4. Module Use	**No modules attached** to the KIBO body	**No correspondence** or **only wheels/motors attached**	**At least one module with correspondence** to the program (in addition to wheels and motors)	**All attached modules are purposeful** and correspond to the constructed program	All attached modules are purposeful **and activated purposefully** when program is running
A5. Data	**No use of sound recorder**	**Sound recorder used incorrectly** (no sound)	**One successfully recorded sound** with sound recorder	**2+ successfully recorded sounds** with sound recorder	**Subroutines** used correctly
B. Project Design Elements					
B1. Sequencing	**3 or fewer blocks** used	**4-5 blocks** used	**6-9 blocks** used	**10-14 blocks** used	**15+ blocks** used
B2. Block Variety	No program or **only Begin/End blocks** used	**1 type** of block used	**2-3 different types** of blocks used	**4-5 different types** of blocks used	**6 types** of blocks used (blue, yellow, orange, gray, lavender, purple)
B3. Robot Customization	**No customization or decorations**	Use of existing KIBO extension modules (ex. platform) as decoration; **no personalized customization**	**1 piece of decoration** added; simple customization	**Personalized arts & crafts and/or building materials** attached to robot	**Highly** personalized arts & crafts and/or building materials **securely attached** to robot
B4. Setting	**No other project elements** added	**1 other project element** added (apart from robot/program)	**2 other project elements** added (apart from robot/program)	**3 other project elements** added (apart from robot/program)	**4+ other project elements** added (apart from robot/program)
B5. Coordination	**No background music or dance, Wait for Clap block, or use of multiple robots**	Used **Wait for Clap block as the final block** before End; no synchronized music or dance	Used Wait for Clap block in **middle of the program**; background music or dance does not fully match program	**Used background music/dance strategically** to match program	**Multiple KIBO robots running synchronously** with background music or dance

Step-by-Step Scoring Guide

Step 1: Add up all of the A scores: _____ out of 20 Then multiply by 1.5: _____ **out of 30**	**Step 2:** Add up all of the B scores: _____ **out of 20**	**Step 3:** Add scores from Steps 1 and 2: _____ **out of 50**	**Step 4:** Identify project's level of complexity: ☐ **Budding (0-9)** ☐ **Developing (10-19)** ☐ **Proficient (20-29)** ☐ **Advanced (30-39)** ☐ **Distinguished (40-50)**

Figure 2. Second Iteration of the KIBO Project Rubric with Analytic and Holistic Scales

This revised rubric was tested with *N* = 50 KIBO projects that were documented using videos, pictures, and/or written descriptions. In order to include the project in our analytic sample, both the block program and the physical KIBO robot needed to be visible. To find projects that fit this inclusion criterion, we used KIBO project videos from earlier research projects and professional development trainings collected by the research team over a number of years, as well as projects that were publicly shared online. This convenience sample included KIBO projects created by young children, as well as adult educators and practitioners. No names were associated with projects; the only recorded characteristic was whether the project was created by an adult or child. The final analytic sample consisted of 25 adult-created and 25 child-created projects.

This revised rubric went through a similar process of iteration and testing. The same team of KIBO experts provided feedback, resulting in minor wording adjustments. To test the rubric, one researcher (Rater 1) scored all 50 projects and trained a second researcher (Rater 2) on the rubric, who also scored all 50 projects. The two raters participated in four rounds of check-ins (after 10, 25, 37, and 50 projects were scored) to assess inter-rater reliability, review scoring discrepancies, and accordingly revise the wording of rubric criteria for better clarity. Multiple inter-rater reliability measures were computed at every check-in. In addition to percentage agreement, weighted kappas using linear weights for the ten scoring constructs and the overall project score were computed. Weighted kappas were used instead of intra-class correlation or other measures because there were two independent raters for the entire dataset, and all 11 variables were treated as ordinal scales (Cohen, 1968; Fleiss & Cohen, 1973).

The next section details the findings from the two phases of rubric testing. Findings include descriptive from both analytic samples (N = 123 projects for the first rubric and N = 50 projects for the second rubric), as well as results from inter-rater reliability analyses for the second analytic sample. Three examples of KIBO projects are also included to demonstrate the rubric's face validity and showcase how the projects' varying levels of complexity were reflected in their KIBO Project Rubric scores. Because of the broad inclusion criteria for projects and the many variables that play a role in the design process of KIBO projects (e.g., nature of the robotics activity, time and space allocated for participants, type of KIBO kit, arts and crafts materials available, etc.), no between-participant nor adult-child comparative analyses were conducted. All performed analyses were done using SPSS Statistics Version 27.

Results

Descriptives

For the first version of the KIBO Project Rubric, the average project score was 2.41 (SD = 1.11) out of a possible 5 points. The overall complexity of projects varied: 39 students' projects (31.7%) were classified as Budding, 14 (11.4%) as Developing, 51 (41.5%) as Proficient, 18 (14.6%) as Advanced, and one (0.8%) as Distinguished. The majority of students did not use a repeat loop (60.5%) or a conditional statement (89.1%) in their KIBO programs. Further syntax analysis indicated that of the 58 students who used a repeat loop, 24 students (41.4%) used it correctly. Of the 16 students who used a conditional statement, 7 of them (43.4%) used it correctly. Over a quarter of students' final programs (35.4%) were syntactically incorrect, almost all of which exhibited errors with repeat loops, conditionals, or their corresponding parameters.

For the second version of the KIBO Project Rubric, the average project score was 24.42 (*SD* = 6.56; range = 11-39) for Rater 1 and 23.91 (*SD* = 5.80; range = 10.5-38) for Rater 2 out of a possible 50 points. The average deviance between the two raters' numerical scores was 1.93 points (*SD* = 1.58). Deviance ranged from 0-7 points. For Rater 1, 15 projects were classified as Developing, 26 projects as Proficient, and 9 projects as Advanced. For Rater 2, 11 projects were classified as Developing, 32 projects as Proficient, and 7 projects as Advanced. None of the 50 projects received a Budding or Distinguished score. Independent samples *t*-tests and a Chi-square analysis indicated no significant difference ($p > .05$) between Rater 1 and 2 in projects' overall numerical scores, sub-scores for Programming Concepts and Project Design Elements, or overall level of complexity. Figure 3 displays the histogram for all 100 scores from both raters, showing a normal distribution.

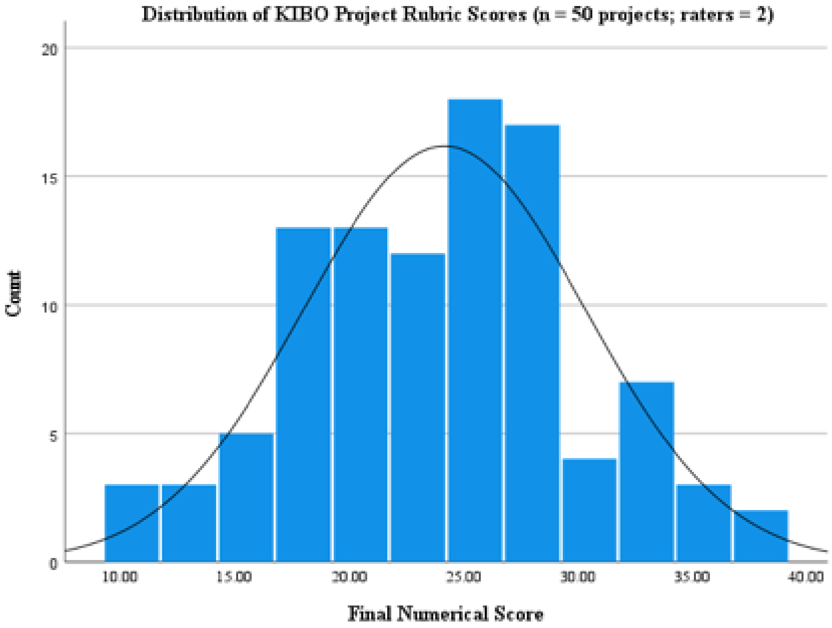


Figure 3. Normal Distribution of KIBO Project Rubric Scores

Reliability

The reliability findings focus on the second iteration of the rubric. Table 3 displays the inter-rater reliability at each of the four rounds of check-ins between the two raters for all ten scoring criteria (A1, A2, A3, A4, A5, B1, B2, B3, B4, and B5). Percentage agreement ranged from 73-77%. Weighted kappa's using linear weights ranged from .802-.838 ($p < .001$), which indicates statistically significant agreement between the two raters. The strength of agreement can be classified as strong agreement beyond chance (Cicchetti & Allison, 1971; Fleiss, Levin & Paik, 2003).

Table 3.
Inter-Rater Reliability for Overall KIBO Project Rubric Scores (N = 50 projects)

Round of Analysis	Percentage Agreement (%)	Weighted Kappa Using Linear Weights (κ_w)
Round 1 (after 10 projects)	76.0	.805, $p < .001$
Round 2 (after 25 projects)	74.3	.802, $p < .001$
Round 3 (after 37 projects)	72.7	.813, $p < .001$
Round 4 (after 50 projects)	77.4	.838, $p < .001$

When each of the ten scoring criteria were analyzed separately for inter-rater reliability, it was evident that some criteria were easier to agree upon than others. Table 4 depicts the weighted kappas using linear weights for each criterion. Block Variety (B2) and Setting (B4), in particular, had poor agreement between the two raters, $\kappa_w < .4$. Module Use (A4) and Coordination (B5) had fair inter-rater agreement, $\kappa_w < .75$. The remaining six criteria had strong inter-rater agreement, $\kappa_w > .75$.

Table 4.
Inter-Rater Reliability for Individual KIBO Project Rubric Criteria (N = 50 projects)

Rubric Criterion	Weighted Kappa Using Linear Weights (κ_w)
A1. Syntactical Accuracy	.851, $p < .001$
A2. Repeats	.905, $p < .001$
A3. Conditionals	.904, $p < .001$
A4. Module Use	.593, $p < .001$
A5. Data	.814, $p < .001$
B1. Sequencing	.935, $p < .001$
B2. Block Variety	.118, $p = .077$
B3. Robot Customization	.764, $p < .001$
B4. Setting	.267, $p < .001$
B5. Coordination	.671, $p < .001$

Percentage agreement and weighted kappa were also computed for the projects' overall level of complexity (Budding, Developing, Proficient, Advanced, and Distinguished). Percentage agreement for the 50 projects was 84%. The eight projects with varying levels from the two raters did not differ by more than one level (e.g., Developing versus Proficient). Weighted kappa using linear weights (κ_w) was .757, $p < .001$, which indicates statistically significant and strong agreement between the two raters (Cicchetti & Allison, 1971; Fleiss, Levin & Paik, 2003).

Validity

The KIBO Project Rubric has adequate face validity, which was established from adapting the rubric criteria from assessments of other coding and robotic technologies for children (e.g., Scratch, ScratchJr, Bee-Bot, etc.). Feedback from the five KIBO experts also supported rubric improvement, particularly in terms of defining key programming concepts. For instance, experts' questions included, “How is correspondence defined?”, “What does it mean for all attached modules to be purposeful?”, and “I think it will be helpful to mention that the block types can be distinguished by color.” Expert review also contributed to content validity of the rubric, which was further enhanced with the additional Project Design Elements set of criteria. To demonstrate the rubric’s face validity, three illustrative examples of children’s KIBO robotics projects are presented.

Project Example 1. Figure 4 displays a kindergarten classroom implementing a KIBO robotics curriculum, in which two students were working together on their final KIBO project. Their program reads, “Begin, Sing, Beep, Shake, Wait for Clap, Red Light On, End”, and affixed to their robot are a lightbulb and sound sensor. The students created a syntactically correct and functional program (A1. 4 points) but did not use any repeats (A2. 0 points) nor conditionals (A3. 0 points). All of the modules attached to the robot—the wheels and motors, lightbulb, and sound sensor—are used and activated purposefully with specific correspondence to the Shake, Red Light On, and Wait for Clap blocks, respectively (A4. 4 points). There is no evidence of data storage in this program (A5. 0 points). Thus, the total sub-score for Programming Concepts is 8 points; when weighted 1.5 times, the sub-score is 12 points. When examining the Project Design Elements of this project, there are a total of seven blocks in this program (B1. 2 points), which include four different types of blocks: orange Sound, blue Motion, purple Wait for Clap, and yellow Light blocks (B2. 3 points). There are no decorations secured to the robot (B3. 0 points) nor other aesthetic elements surrounding the project (B4. 0 points). There is some evidence of coordination because of the placement of the Wait for Clap block in the middle of the KIBO program (B5. 2 points). The total sub-score for Project Design Elements is 7 points, bringing the total numerical score to 19. Thus, this project displays a Developing level of complexity.

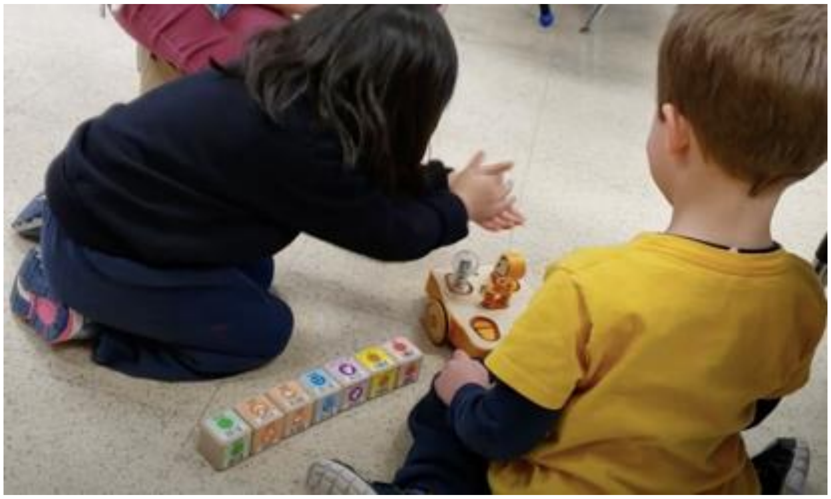


Figure 4. KIBO Robotics Project Example 1

Project Example 2. Figure 5 displays another kindergarten classroom that utilized the storybook *Brown Bear, Brown Bear, What Do You See?* by Bill Martin, Jr. and Eric Carle as the theme for their KIBO projects. A kindergarten student designed and programmed his KIBO robot to move through the taped illustrations, which depict the brown bear going to a farm and seeing a white dog. The KIBO program reads, "Begin, White Light On, Repeat, Three, Forward, Turn Left, End Repeat, End". Affixed to the robot are wheels and motors, as well as the lightbulb module and art platform that holds the student's personalized brown bear decorations. The program has no syntactical errors (A1. 4 points) and successfully utilizes a repeat loop with a number parameter (A2. 2 points). No conditionals are used (A3. 0 points). All attached modules are used purposefully, but because the program does not utilize any sensors, there is no purposeful activation (A4. 3 points). There is no evidence of data storage in this KIBO program (A5. 0 points). The total sub-score for Programming Concepts is 9 points; when multiplied by 1.5, the weighted sub-score is 13.5 points. With respect to the Project Design Elements, there are a total of eight blocks in this program (B1. 2 points) of three different types: yellow Light, gray Repeat, and blue Motion blocks (B2. 2 points). The robot's decorations are highly personalized, using various kinds of arts and crafts materials (B3. 4 points). The project's setting includes three distinct pieces of artwork taped to the floor, which are integral components of the book-themed project (B4. 3 points). Finally, there is no evidence of coordination in the form of multiple robots moving simultaneously, use of background music, or the Wait for Clap block (B5. 0 points). The total sub-score for Project Design Elements is 11 points, bringing the total numerical score to 24.5. Thus, this project receives a Proficient level of complexity.

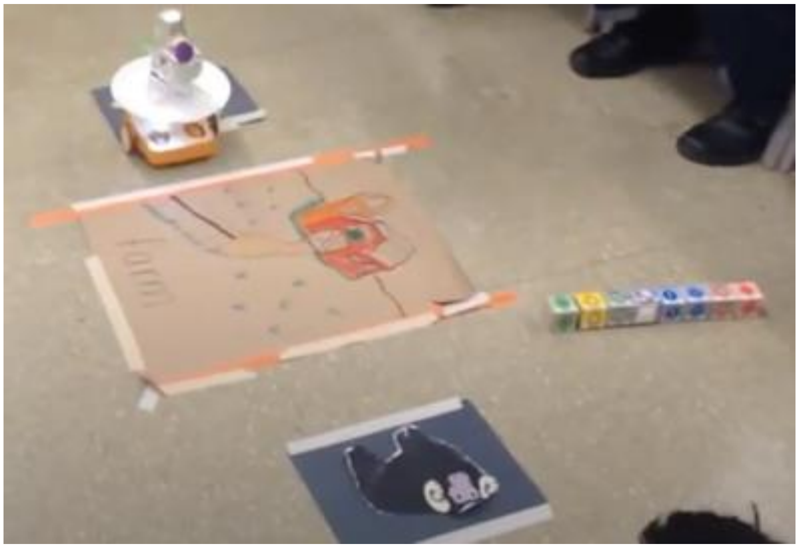


Figure 5. KIBO Robotics Project Example 2

Project Example 3. Figure 6 displays a KIBO project created by a second-grade student. The constructed block sequence, which reads "Begin, Repeat, Three, If, Light, White Light On, Red Light On, Blue Light On, End If, If, Dark, Wait for Clap, Wait for Clap, Forward, End If, End Repeat, End", is syntactically correct (A1. 4 points). There are two consecutive conditional statements inside of a repeat loop, demonstrating the child's ability to sequence a nested statement (A2. 4 points and A3. 4 points). Although not all the modules are displayed in this photo, supporting video documentation indicated that the child had appropriately affixed the wheels and motors, light sensor, lightbulb, and sound sensor to her robot. However, when the child was testing the KIBO program, she had trouble with triggering the light sensor with the flashlight; thus, the light sensor was not successfully activated (A4. 3 points). There is no use of data storage in this project (A5. 0 points). With respect to the project's design elements, the child used more than 15 blocks (B1. 4 points) of 5 different kinds (B2. 3 points). There is no customized decoration apart from the existing platform extension (B3. 1 point). The flashlight, which was used to trigger the light sensor, is considered part of the setting (B4. 1 point). Finally, the Wait for Clap blocks are placed intentionally inside one of the conditional statements (B5. 3 points). Altogether, this project receives 15 points for Programming Concepts (weighted to 22.5 points) and 11 points for Project Design Elements, which brings the total number to 33.5 points. Overall, this project displays an Advanced level of complexity.

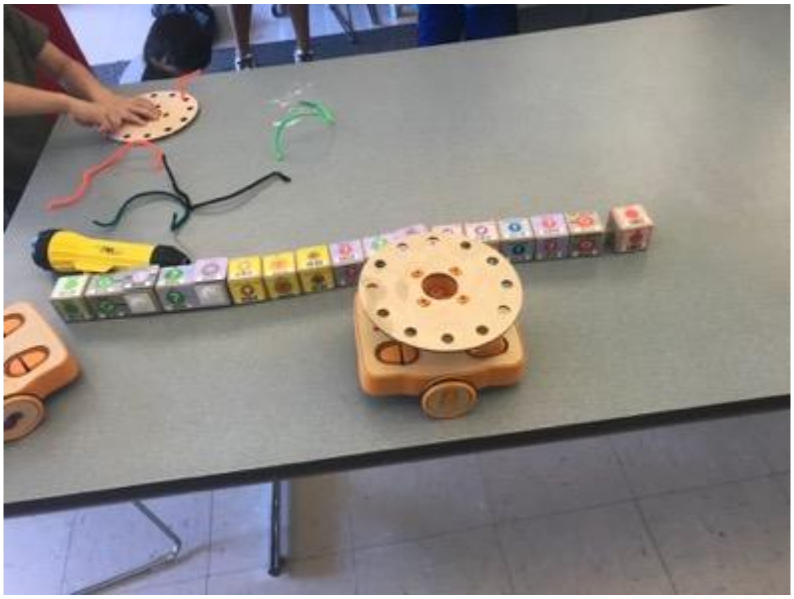


Figure 6. KIBO Robotics Project Example 3

The picture and video documentation of these example projects were useful in determining each project's level of complexity. From the pictures alone, it is evident that the second example contains more aesthetic project elements than the first example. This second example also utilized a repeat loop, which is a more advanced programming concept. Thus, it is not surprising that this second project received a higher numerical score and overall level of project complexity in comparison to the first project. However, one area in which the first example surpassed the second is the Module Use criterion. The use of the sound sensor and Wait for Clap block in the first project demonstrates an advanced understanding of KIBO sensors, which was not evident in the second project that only utilized the lightbulb module. Although barely decorated, the third example project displayed skillful use of repeat loops and conditional statements, which contributed to its high score. As shown in these three examples, there are many different components of KIBO robotics projects, which were captured by the various criteria in our KIBO Project Rubric.

Discussion

Early childhood robotics learning has become an increasing focus of early computer science and STEAM education efforts. The screen-free KIBO robotics platform is one such technological tool that engages young children in foundational programming concepts while also promoting creativity, self-expression, and design thinking. Although there are various assessments for children's KIBO learning (DevTech Research Group, 2019), these assessments fall under the category of multiple-choice questionnaires, interview protocols, and design scenarios. There are no KIBO rubrics for assessing young children's robotics projects with adequate psychometric properties. This paper filled this gap by presenting the multi-phase development and testing process of the KIBO Project Rubric, a project-based assessment tool for the KIBO robotics platform. Using a total of 173 KIBO projects, 123 of which were recorded in student design journals and 50 of which were recorded using picture and video documentation, the psychometric properties of the KIBO Project Rubric were explored, and the rubric was iteratively improved in its design and format.

This discussion primarily focuses on the second version of the KIBO Project Rubric, as this rubric provides a more comprehensive examination of KIBO project artifacts. However, the first rubric, which consisted of a single 5-point scale, might still be a suitable assessment tool if the goal is to focus specifically on the KIBO programming language (i.e., the block program). In that case, this rubric would be relatively quick to administer and would foster insight into students' depth of understanding and application of programming concepts in their constructed codes. For instance, findings indicated that children were unlikely to utilize repeats and conditionals in their KIBO projects, even if they engaged in those topics in a full-length KIBO curriculum. The majority of children who did use these advanced KIBO blocks exhibited syntax errors (e.g., missing a Begin or End Repeat/If block or the proper corresponding parameter). There may be a number of reasons for these errors, such as students being unable to record their programs accurately using the KIBO stickers, not having enough time to check over their work, or not fully understanding how these advanced concepts are used. If the latter reason is true, then this is further reason to promote the use of multiple assessments to understand the full extent of children's programming knowledge. Prior research has showed that kindergarten, first, and second grade students performed equally well on advanced programming KIBO Solve-Its, which included questions about repeats and conditionals (Sullivan & Bers, 2015). Perhaps it may be that children can correctly answer questions about these advanced programming concepts but do not exhibit this knowledge when given the opportunity to apply these concepts in their projects. Further research is needed in this area to explore how children may exhibit their programming knowledge using various assessment methods.

The rubric development and testing process reemphasized the notion that programming is only one aspect of a robotics project. Because of the customizable art platform and the various aesthetic elements that can be added to make KIBO robotic creations come alive, there was a need to expand the KIBO Project Rubric to include these design elements. Thus, if the goal is to examine KIBO robotics projects in their entirety, the second iteration of the KIBO Project Rubric is a more suitable assessment tool. Overall, this rubric demonstrated good psychometric properties. Multiple forms of validity (e.g., construct, content, and face validity) were investigated. Further rubric validation is required, perhaps by inviting teachers to use the rubric to assess students' KIBO projects and comparing their ratings to researchers' ratings of the same projects.

Findings from inter-rater reliability analyses indicated strong agreement beyond chance between the two raters. However, there were four criteria with low-to-medium inter-rater agreement: Block Variety (B2), Setting (B4), Module Use (A4), and Coordination (B5). What might be the source of these discrepancies? About halfway through scoring, it became known that the two raters disagreed on whether the Begin and End blocks should be considered as a type of block. It was then clarified that the Begin and End blocks do not count for this criterion. In addition, Setting, Module Use, and Coordination were sometimes difficult to assess, depending on the quality of picture or video documentation. Perhaps these discrepancies may be resolved if KIBO projects are documented thoroughly to display all aspects of the program and the robot. Video documentation, in particular, might be more suitable, especially if the project contains background music, or if the child is narrating their project idea in the background. To remedy some of the scoring discrepancies and further improve the reliability of the KIBO Project Rubric, examples for each 0-4 marker for each of the ten scoring criteria were added. This third and current version of the KIBO Project Rubric (available online at http://bit.ly/kibo-project-rubric) will be retested in future work and reexamined for its psychometric properties. Guidance on documentation should also be provided to assessors, specifying that project documentation should include the KIBO block sequence, physical robot in its decorated form, and a video of the robot in action.

There are several important points to note about the rubric's final project scores. One is the positive framing for the names of the five level categories. Aligned with the principles of strengths-based education (Lopez & Louis, 2009), the level of project complexity obtained from the KIBO Project Rubric is meant to highlight the strengths of project creators' efforts and achievement, rather than position any misconceptions in their learning as deficits. Positive framing also serves to position learning with the KIBO robotics kit as a developmental activity. A person (child or adult) who is introduced to KIBO for the very first time, even if older by age, may not necessarily create a "proficient" project. By using the terms "budding" or "developing", the rubric acknowledges that each project creator is growing their programming skills and with more experience and exposure, they may have the opportunity to produce more complex KIBO projects.

It is also essential to note that the final score does not indicate overall level of programming mastery. Rather, the score provides an estimated level of mastery *as exhibited in this particular project*, which means that projects might be limited by factors outside of their control. For example, children who are working with KIBO-10 (an introductory kit containing the 10 basic programming blocks) are likely to create projects that are less complex than children working with KIBO-21 (a more comprehensive kit with additional advanced blocks and sensors). Unless children are prompted to demonstrate their most advanced programming skills in their projects, as well as provided with unlimited time and resources for building and decorating their robots, children would not be expected to display the full extent of their knowledge in a single project. Thus, a limitation of a project-based assessment is that it is only one way of understanding children's KIBO knowledge. Another limitation is that half of the second analytic sample examined adult-created projects. Although no adult-child comparative analyses were presented in this paper, future work may explore the level of complexity exhibited in adults' KIBO projects in comparison to children's KIBO projects, as well as projects created by children of varying age and ability levels.

Conclusions

This paper presents the multi-phase development and testing for a robotics project rubric. Key lessons were learned in this process, such as documenting both process and outcome of robotics projects, acknowledging possible subjectivity and ambiguity in scoring project artifacts, and emphasizing projects' creative intent as much as exhibited content knowledge. The findings of this paper, as well as these lessons learned, can be used to inform the development of rubrics for other robotics platforms for young children. For example, robotics projects with Bee-Bot® or Code-a-Pillar™ also involve both programming and aesthetic design elements that could be assessed to shed insight into children's learning. In addition, the process through which the KIBO

Project Rubric was developed and tested reveals the iterative nature of rubric design. This method is essential because any artifact with practical use in both research and educational settings should be developed iteratively with feedback from researchers and practitioners. Future work for the KIBO Project Rubric will continue to seek stakeholder feedback, not only for the rubric, but also for user training and calibration.

References

Albo-Canals, J., Martelo, A. B., Relkin, E., Hannon, D., Heerink, M., Heinemann, M., ... & Bers, M. U. (2018). A Pilot Study of the KIBO Robot in Children with Severe ASD. *International Journal of Social Robotics*, 1-13. https://doi.org/10.1007/s12369-018-0479-2.

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. https://doi.org/10.1145/1929887.1929905.

Benitti, F. B. V. (2012). Exploring the educational potential of robotics in schools: A systematic review. *Computers & Education*, *58*(3), 978–988. https://doi.org/10.1016/j.compedu.2011.10.006.

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

Bers, M. (2019). Coding as another language: A pedagogical approach for teaching computer science in early childhood. *Journal of Computers in Education*, *6*(4), 499–528. https://doi.org/10.1007/s40692-019-00147-3.

Bers, M. (2020). *Coding as a Playground: Programming and Computational Thinking in the Early Childhood Classroom, Second Edition*. Routledge Press.

Bers, M., González-González, C., & Armas–Torres, M. B. (2019). Coding as a playground: Promoting positive learning experiences in childhood classrooms. Computers & Education, 138, 130-145. https://doi.org/10.1016/j.compedu.2019.04.013.

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

Brennan, K., Haduong, P., & Veno, E. (2020). *Assessing Creativity in Computing Classrooms.* Creative Computing Lab.

Brennan, K., & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking*.

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

Cicchetti, D. V., & Allison, T. (1971). A New Procedure for Assessing Reliability of Scoring Eeg Sleep Recordings. *American Journal of EEG Technology*, 11(3). https://doi.org/10.1080/00029238.1971.11080840.

Clements, D. H., & Gullo, D. F. (1984). Effects of computer programming on young children's cognition. *Journal of Educational Psychology, 76,* 1051-1058. https://doi.org/10.1037/0022-0663.76.6.1051.

Cohen, J. (1968). Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin, 70*(4), 213–220. https://doi.org/10.1037/h0026256.

DevTech Research Group (2018). General Assessment Templates. https://sites.tufts.edu/devtech/files/2018/03/GeneralAssessments.pdf

DevTech Research Group (2019). A Guide to Replicating a KIBO Robotics Study. http://sites.tufts.edu/devtech/research/kibo-robot/

Elkin, M., Sullivan, A., & Bers, M. (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.

Fleiss, J. L., & Cohen, J. (1973). The Equivalence of Weighted Kappa and the Intraclass Correlation Coefficient as Measures of Reliability. *Educational and Psychological Measurement, 33*(3), 613-619. https://doi.org/10.1177/001316447303300309.

Fleiss, J. L., Paik, M. C. & Levin, B. (2003). *Statistical Methods for Rates and Proportions*. John Wiley Son Inc

Grover, S. (2017). Assessing Algorithmic and Computational Thinking in K-12: Lessons from a Middle School Classroom. In *Emerging Research, Practice, and Policy on Computational Thinking* (p. 269-288). Springer International.

Grover, S. (2020). Designing an Assessment for Introductory Programming Concepts in Middle School Computer Science. In *Proceedings of 51st ACM Technical Symposium on Computer Science Education (SIGCSE'20)*, https://doi.org/10.1145/3328778.3366896.

Hassenfeld, Z., Govind, M., De Ruiter, L., & Bers, M. (2020). If You Can Program, You Can Write: Learning Introductory Programming Across Literacy Levels. *Journal of Information Technology Education: Research*, *19*, 065–085. https://doi.org/10.28945/4509.

Horn, M. & Bers, M. (2019). Tangible Computing. In S.A. Fincher & A.V. Robins (Eds.), *The Cambridge Handbook of Computing Education Research.* Cambridge University Press.

Jurado, E., Fonseca, D., Coderch, J., & Canaleta, X. (2020). Social STEAM Learning at an Early Age with Robotic Platforms: A Case Study in Four Schools in Spain. *Sensors, 20.* https://doi.org/10.3390/s20133698.

Kafai, Y., & Burke, Q. (2014). *Connected Code: Why Children Need to Learn Programming*. The MIT Press.

Lopez, S. J., & Louis, M. C. (2009). The Principles of Strengths-Based Education. *Journal of College and Character*, *10*(4). https://doi.org/10.2202/1940-1639.1041.

Meacham, S., & Atwood-Blaine, D. (2018). Early childhood robotics with inspirations from Reggio Emilia educators. *Science & Children*, *56*(3), 57-62.

O'Malley, C., & Fraser, D. S. (2004). *Literature Review in Learning with Tangible Technologies*. 53. A NESTA Futurelab Research report - report 12.

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

Papert, S., & Harel, I. (1991). *Constructionism*. Ablex Publishing.

Portelance, D. J., & Bers, M. (2015). Code and tell: Assessing young children's learning of computational thinking using peer video interviews with ScratchJr. *Proceedings of the 14th International Conference on Interaction Design and Children - IDC '15*, 271–274. https://doi.org/10.1145/2771839.2771894.

Pugnali, A., Sullivan, A., & Bers, M. (2017). The impact of user interface on young children's computational thinking. *Journal of Information Technology Education: Innovations in Practice, 16,* 172-193. https://doi.org/10.28945/3768.

Relkin, E. & Bers, M. (2019). Designing an Assessment of Computational Thinking Abilities for Young Children. In L.E. Cohen & S. Waite-Stupiansky (Eds.), *STEM for Early Childhood Learners: How Science, Technology, Engineering and Mathematics Strengthen Learning* (pp. 85-98). Routledge.

Relkin, E., de Ruiter, L., & Bers, M. (2020). TechCheck: Development and Validation of an Unplugged Assessment of Computational Thinking in Early Childhood Education. *Journal of Science Education and Technology*, *29*(4), 482–498. https://doi.org/10.1007/s10956-020-09831-x.

Resnick, M. (2007). All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten. Presented at Creativity & Cognition conference. https://web.media.mit.edu/~mres/papers/CC2007-handout.pdf

Salac, J., & Franklin, D. (2020). If They Build It, Will They Understand It? Exploring the Relationship between Student Code and Performance. *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*, 473–479. https://doi.org/10.1145/3341525.3387379.

Seiter, L. & Foreman, B. (2013). Modeling the learning progressions of computational thinking of primary grade students. *Proceedings of the ninth annual international ACM conference on International computing education research.* https://doi.org/10.1145/2493394.2493403.

Strawhacker, A. & Bers, M. (2015). "I want my robot to look for food": Comparing children's programming comprehension using tangible, graphical, and hybrid user interfaces. *International Journal of Technology and Design Education, 25*(3), 293-319. https://doi.org/10.1007/s10798-014-9287-7

Strawhacker, A., & Bers, M. (2019). What They Learn When They Learn Coding: Investigating cognitive domains and computer programming knowledge in young children. *Educational Technology Research and Development, 67*(3), 541-575. https://doi.org/10.1007/s11423-018-9622-x.

Sullivan, A. (2019). *Breaking the STEM Stereotype: Reaching Girls in Early Childhood.* Rowman & Littlefield.

Sullivan, A., & Bers, M. (2015). Robotics in the early childhood classroom: Learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade. *International Journal of Technology and Design Education, 26,* 3-20. https://doi.org/10.1007/s10798-015-9304-5

Sullivan, A., Bers, M., Mihm, C. (2017). Imagining, Playing, & Coding with KIBO: Using KIBO Robotics to Foster Computational Thinking in Young Children. In *Proceedings of the International Conference on Computational Thinking Education.* Wanchai

Sullivan, A., Elkin, M., & Bers, M. (2015). KIBO Robot Demo: Engaging young children in programming and engineering: *Proceedings of the 14th International Conference on Interaction Design and Children (IDC '15)*, Medford, MA, New York, NY: ACM

Wing, J. M. (2006, March). Computational Thinking. *CACM Viewpoint*, 33-35. http://www.cs.cmu.edu/afs/cs/usr/wing/www/publications/Wing06.pdf

Wohl, B., Porter, B., Clinch, S. (2015). Teaching computer science to 5–7-year-olds: An initial study with Scratch, Cubelets and unplugged computing. *Proceedings of the Workshop in Primary and Secondary Computing Education,* 55–60. https://doi.org/10.1145/2818314.2818340.

Yu, J., & Roque, R. (2018). A survey of computational kits for young children. *Proceedings of the 17th ACM Conference on Interaction Design and Children - IDC '18*, 289–299. https://doi.org/10.1145/3202185.3202738