TechCheck-K: A Measure of Computational Thinking for Kindergarten Children

Emily Relkin
Eliot Pearson Dept. of Child Study and Human Development
Tufts University
Medford, USA
Emily.Relkin@tufts.edu

Marina Bers
Eliot Pearson Dept. of Child Study and Human Development
Tufts University
Medford, USA
Marina.Bers@tufts.edu

***Abstract*— In this pilot study we created and evaluated a new version of the *TechCheck* Computational Thinking (CT) assessment designed specifically for kindergarten students. The original version of *TechCheck* was validated in first and second graders and consists of 15 multiple choice unplugged challenges drawn from six CT domains. The kindergarten version, called *TechCheck-K,* consists of similar unplugged challenges but reduces the number of response choices from four to three and uses exclusively non-verbal material in the responses to accommodate younger, pre-literate children. We administered *TechCheck-K* to *N*= 89 kindergarten students from four schools, including 45 boys and 44 girls with no coding experience. Participants completed the assessment in an average of 23.43 minutes. *TechCheck-K* scores were normally distributed (*M*= 8.26, *SD*= 2.33). Mean scores were numerically higher in boys (*M*= 8.71, *SD*= 2.69) than girls (*M*= 8.05, *SD*= 2.07). The difference between genders was not significant. The pattern of correct responses to the 15 *TechCheck-K* items correlated (*r* = .76) with the pattern observed in first and second graders on the original *TechCheck*, providing evidence that comparable CT constructs are measured by the two versions of the instrument. Results of this pilot study indicate that *TechCheck-K* has acceptable characteristics for assessment of CT in kindergarten children.**

Keywords—Computational thinking, Assessment, Unplugged, Computer Science, Early Childhood Education

I. INTRODUCTION

Computer science (CS) is becoming increasingly ubiquitous in early childhood education, including kindergarten classrooms [1-3]. A desirable outcome of learning to code and gaining technological fluency is the acquisition of computational thinking (CT) skills. Although the concept of CT is still evolving it can be defined as the set of thought processes that facilitate framing and solving problems using computers and other technologies [4,5]. CT assessment is important in CS education for documenting learning progress, measuring lesson effectiveness, assisting in curriculum development and helping identify students in need of greater assistance or enrichment [6,7]. There is a recognized need for well-validated, age-appropriate CT assessments for young children for use in educational and research settings [8,9].

CT assessments have traditionally been based on performance on coding challenges. For example, Bers et al., [10] assessed kindergarteners (ages 4 to 6) using a tangible and graphical programming language called CHERP and LEGO® robotics. Children were tasked with programming their robot to dance. The researchers rated debugging, correspondence, sequencing, and control flow by using project-based assessment to score children's projects on a Likert scale. While this is a viable approach to assessing CT, the use of programming exercises for assessment purposes requires that children have at least rudimentary knowledge of coding, which excludes some children from the assessment. This approach also risks conflating CT with coding abilities [11]. To circumvent these issues, Hoffer, et al., [12] proposed assessing CT skills in K-12 students by observing algorithmic, problem solving, and creativity outcomes in pairs of children constructing structures with cubes. This approach is difficult to standardize and has not gained wide acceptance. In an alternative approach, Román-González et al., created an CT assessment called the "Computational Thinking test" (CTt) [13,14] that employs unplugged challenges analogous to the unplugged activities that are sometimes used to teach computer science principles without the use of a computer. The CTt does not require students to have knowledge of programming languages or coding. While the use of unplugged challenges represents an innovative approach to CT assessment, the CTt was designed for middle school and older students and is not suitable for use with children in kindergarten.

In 2020, two unplugged CT assessments for elementary school age children were published: *TechCheck* [15] and the BCTt [16]. The BCTt is an unplugged CT assessment based on the CTt that was designed for elementary school children and performed best in children in 1st- 2nd grade. The BCTt probes the CT constructs of sequences, loops (simple and nested), conditionals (if-then, if-then-else, and while). The estimated administration time of the BCTt is 40 minutes, which is less than ideal for kindergarten students.

TechCheck is a 15-item unplugged CT assessment based on Bers' Seven Powerful Ideas of CS [17]. The original version of *TechCheck* was designed for children ages 5-9 years and employs unplugged challenges that are analogous to those encountered in the course of computer programming. *TechCheck* can be administered to large numbers of students simultaneously in under a half hour and does not require prior knowledge of coding to complete. The CT domains probed by *TechCheck* correspond to six of Bers' Seven Powerful Ideas of CS, including Algorithms, Modularity, Control Structures, Representation, Hardware and Software (See Appendix). Correct answers on each of the fifteen multiple choice questions are awarded one point, and there are no penalties for incorrect responses. Two practice questions are included at the beginning of the assessment to familiarize students with the format but are not included in the scoring. All questions must be answered to complete the assessment.

TechCheck was validated in a sample of (*N* = 768) students between 5 and 9 years old [15]. *TechCheck* scores were normally distributed and correlated moderately with a previously validated CT assessment tool (*TACTIC-KIBO*). The instrument demonstrated good reliability and validity as measured by classical test theory and item response theory. Relkin et al., subsequently used *TechCheck* in a longitudinal study of first and second graders that compared students who received the CAL coding curriculum over 7 weeks to age-matched controls who participated in regular classroom activities over the same period of time [18]. *TechCheck* scores improved significantly in those exposed to the coding curriculum compared to their non-coding counterparts. CT improvements after seven weeks of the coding curriculum equated to changes seen over 6 months of normal development. Algorithms, modularity, and representation probes showed the most improvements in children who coded. A planned extension of these studies to kindergarten students was delayed by the COVID-19 pandemic.

Assessment of kindergarteners in general can be challenging as it requires special sensitivity to their cognitive, literacy and motor development [19]. Previous research has found that the working memory of kindergarten age children (~5 years old) limits them to holding an average of three items in immediate memory, compared to children in first and second grade (~6-9 years old) who can hold an average of four items [20,21]. This limit could impact kindergartener's performance on multiple choice assessments, especially those with more than three choices. There is currently relatively limited information about the domains of CT that kindergarten children can master. Some researchers have reported that children in kindergarten have a very limited ability of understanding conditionals, and that the capacity to understand *if-then* constructs typically develops between the ages of 6 and 12 years [22,23]. However, other studies have found that performance on tasks related to conditionals or loops are not limited to specific ages [24, 25]. These and other considerations speak to the importance of designing a CT assessment that is developmentally appropriate for kindergarteners.

In order to be able to use the previously validated *TechCheck* assessment with children in kindergarten, we created a modified instrument called *TechCheck-K*. The purpose of the present study was to answer the following research questions:

1. Does *TechCheck-K* have utility as an unplugged assessment of computational thinking for kindergarten students?

2. Does *TechCheck-K* show equivalence to the original version of the *TechCheck* assessment validated in first and second graders?

II. METHOD

A. *TechCheck-K* Development

TechCheck-K is a screening assessment designed to probe six of the CT constructs outlined by Bers [17], as powerful ideas of CS. These constructs are Algorithms, Modularity, Design Process, Debugging, Control Structures, Hardware/Software (see Appendix for further information). Design Process is the seventh powerful idea outlined by Bers [17] but it was not included in the assessment because it is an inherently iterative and open-ended process that does not lend itself to *TechCheck-K*'s multiple choice format.

In modifying the assessment for kindergarten students, we began by reviewing the stems of 15 item with the intention of simplifying wording as much as possible. In both the original *TechCheck* and the version for kindergarten students, the stem portion is read out loud by a proctor up to two times. Current versions do not include accommodations for hearing impairment or non-English language speakers.

To reduce the processing demands associated with the original four choice assessment, we systematically eliminated one distractor per question. We used data from the first and second graders responses to identify pairs of distractors with close to the same response probabilities and eliminated one for each question. We followed this procedure in an effort to maintain the overall difficulty and discrimination levels on a par with the original version of *TechCheck*. With one distractor per question eliminated, *TechCheck*-K has three response alternatives per question instead of the four in the original version (see Figure 1).

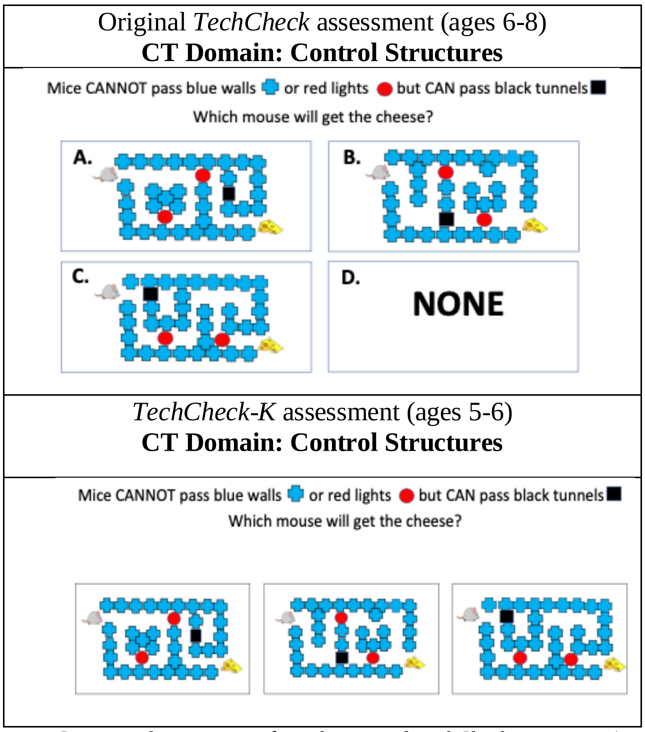


Fig. 1 Corresponding questions from the original *TechCheck* assessment (*top*) and the new *TechCheck-K (bottom)*

We previously observed that some first and second graders had difficulty accurately entering their identifying information at the beginning of the assessment. This led to lengthy preparation times for proctors and/ or additional waiting time for children taking the assessment. For the *TechCheck*-K version, we created a drop-down list of children's names from class rosters to make registration by a proctor easier. Previously, we found that some younger children had trouble using tablets or computers when required to use a mouse or trackpad. In this pilot study, we administered the assessment using Apple iPads. The touchscreen made it easier for children to select their response by touching their choice. Due to the COVID-19 pandemic, proctors tested n=10 children individually over the Zoom platform. In those cases, much the same administration procedures were followed but a parent was present in the room with the child to facilitate the Zoom call and children indicated their choices either by pointing with a mouse or verbally instead of by using a touchscreen.

B. Procedure

The eight proctors participating in this pilot study attended an hour-long training session in which they received instruction on how to administer the assessment. The proctors practiced administration and had to demonstrate their competency on a certification exam before being given clearance to administer the assessment to students. Parental opt-out consent was employed in two of the schools and opt-in consent was carried out in the two other schools. The *TechCheck-K* assessment was administered to children using the Qualtrics survey platform (www.qualtrics.com). Data was collected electronically via the Qualtrics API.

Each administration room was equipped with a screen for projecting the assessment to help orient children to the questions being presented on their iPads. Proctors chose names from a drop-down list on each iPad to register each student for the assessment. Proctors established rapport with students through casual conversation and asked children to provide verbal assent prior to administration. All children gave their assent for participation in this study.

The first two questions on *TechCheck-K* were for practice so children become familiar with the assessment platform and format. These questions did not count towards the total score. By observing their performance with practice questions, proctors could ensure that children were complying with the assessment protocol and could indicate their responses on their iPads. Proctors were instructed to read each stem question out loud up to two times. If children indicated that they did not know an answer they were instructed to guess. If required, a break lasting up to five minutes was permitted during the assessment period.

C. Statistical Analysis

Statistical analyses were conducted in R Version 3.6.1 [26] using R Studio R Studio version 1.2 [27].

III. RESULTS

A. Participants

Participants included 89 kindergarten students from two schools in Virginia and one in Massachusetts, USA. The average age of participating students was *M*= 5.84, *SD* =0.44 with a minimum age of 5 and a maximum age of 6. The majority of participants were White (51.69%). There were roughly equal numbers of boys (*N*=45) and girls (*N*=44) participated. Table 1 summarizes the demographic information for the participants.

TABLE I. DEMOGRAPHICS OF *TECHCHECK-K* PARTICIPANTS

Total N		89
Self- reported age	Mean	5.84
	SD	0.44
	Range	5-6
Gender	Female	44
	Male	45
Race/ethnicity	Black/African American	21 (23.60%)
	Hispanic or Latino/a	5 (5.61%)
	Biracial/ Multiracial	12 (13.48%)
	Asian or Pacific Islander	2 (2.25%)
	White	46 (51.69%)
	American Indian/Native American	1 (1.12%)
	Unknown	2 (2.25%)

B. Descriptives

All children who assented to participate were able to complete the assessment. Administration time averaged 23.43 minutes, *SD*= 11.67. The minimum testing time was 8.88 and the maximum was 44.61. The calculated completion times included break times and the time proctors spent registering students. Actual administration time for the assessment itself was therefore less than the reported average.

There was a normal distribution of *TechCheck-K* scores (*M*= 8.38, SD= 2.41) with a minimum score of 0 out of 15 and maximum score of 14 out of 15 (see Figure 2). The mean among boys (*M*=8.71 *SD*= 2.69) was slightly higher than among girls (*M*=8.05 *SD*= 2.07). However, a Welch two samples t-test showed that there was no statistically significant difference between the genders (*t*=1.31, *df*= 82.4, *p*>.05).

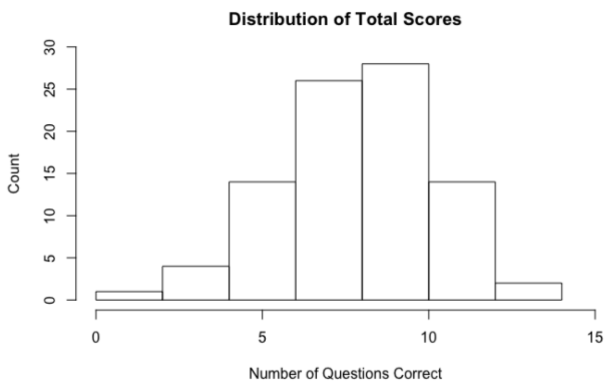


Fig. 2. Distribution of *TechCheck-K* scores among all participants

C. Version Comparisons

Density plots showing the distribution of scores adjusted for the number of participants in each grade are shown in Figure 3. All three grades showed bell-shaped distributions of scores. The lowest peak score was seen in kindergarten students, intermediate in first grade and highest in second grade, consistent with expectations for students in the respective grades. An ANOVA was conducted which showed that there was a significant difference in baseline scores by grade (*F*(2, 854)=133.80, *p* < .001). A Tukey's HSD post hoc test revealed that the difference was only significant between kindergarten and second grade (*p*<.001) and first and second grade (*p*<.001).

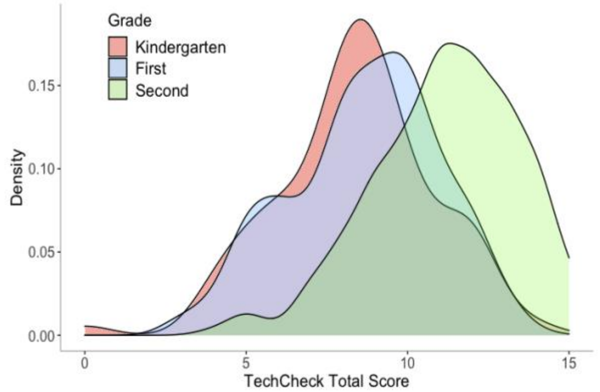


Fig. 3. Density plots of scores on *TechCheck*-K (red, kindergarten) and original *TechCheck* (blue 1st grade; green, 2nd grade). Source of first and second grade data: [18]

The pattern of correct response to the 15 *TechCheck-K* items correlated (r=.76) with the response pattern observed in first (N= 288) and second graders (N= 480) taking the original *TechCheck* [17]. Figure 4 shows the percentage of correct for each question by kindergarten, first, and second grade students respectively. Overall, the pattern of responses to *TechCheck-K* closely followed that seen with the original *TechCheck* in older students.

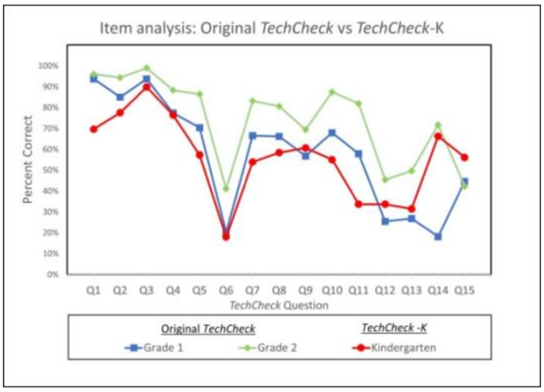


Fig. 4. Percent correct responses by question on *TechCheck* (red, kindergarten) vs. original *TechCheck* (blue 1st grade; green, 2nd grade). Source of 1st and 2nd grade data: [15]

V. DISCUSSION

In this pilot study we field-tested a new version of the CT assessment *TechCheck* that was modified for administration to kindergarten children. In general, the modified version proved engaging to participants and was easy to administer and score. While no student withdrew assent or failed to complete, one student scored zero. Calculated administration times were longer than anticipated owing to inclusion of the time spent on breaks and registering students on their iPads in the calculation of duration of testing. Nevertheless, an average administration time of approximately 24 minutes is acceptable for this age group.

We are encouraged by the assessment's performance in generating a normal distribution of scores in the expected range for kindergarteners. Likewise, it is noteworthy that the percentage of correct responses for each item on *TechCheck-K* closely paralleled that observed with the original version of *TechCheck* in first and second grade students. We would not expect to see a correlation if student responses were predominantly random guesses or if the CT constructs embodied in the original *TechCheck* were not being assessed in the kindergarten version.

TechCheck-K expands the repertoire of instruments available to assess CT in kindergarten children. Its brief, multiple choice format makes it suitable for the level of working memory and literacy skills exhibited by typically developing kindergarteners. Alternative assessments such as the BCTt [16] require longer administration times and have not yet been normed for kindergarten students. In contrast to other CT instruments for kindergarteners such as TACTIC-KIBO [28,29], The Coding Development (CODE) Test 3-6 [30], or The Holistic Computational Thinking Assessment Rubric [31], *TechCheck-K* can be given to children regardless of whether or not they have prior coding experience. This avoids potentially conflating CT with coding skills.

A. Limitations

We originally intended to have the assessment administered to children wearing headphones so that the questions could be read out loud in a uniform manner. However, due to constraints imposed by the survey platform software, we were unable to successfully implement this feature and instead reverted to having proctors read out the questions.

This pilot study does not include a representative sample of all kindergarten students, nor did it include students with neuro-diverse development. The study was not adequately powered to fully assess the effect of factors such as gender and race on the outcomes.

As in the case of the original *TechCheck*, the multiple-choice format of the kindergarten instrument does not lend itself to the creative self-expression and open-ended problem solving that we recognize is a significant part of CT. These limitations should be kept in mind when interpreting the results of CT assessment using *TechCheck*. Other forms of assessment can be used in combination with *TechCheck-K* to get a more complete picture of the child's technology related skills. For example, the PTD checklist [32, 33] measures six positive behaviors (creativity, communication, collaboration, community building, content creation, and choices of conduct) that contribute to positive technological development. Future studies should explore more naturalistic means of measuring CT in young children including valid and reliable observational checklists and game-based assessments. The original *TechCheck* has been translated into several languages besides English and is currently being used in educational and research settings in several countries around the world. We believe *TechCheck-K* can further extend the utility of the instrument to kindergarten students and help shed light on the development of CT skills in young children.

ACKNOWLEDGMENT

We would like to thank our Project Coordinators, Angela de Mik and Megan Bennie for their many contributions to this work. We would also like to thank all of the educators, parents, and students who participated in this study, as well as members of the DevTech research group at Tufts University. This study was generously funded by the Department of Defense Education Activity (DoDEA) grant entitled "Operation: Break the Code for College and Career Readiness." Unique Entity Identifier: "WORLDCL10" and The Department of Education PR/Award Number: U411C190006

REFERENCES

[1] M.U. Bers, Coding as a Playground: Programming and Computational Thinking in the Early Childhood Classroom, 2nd ed. Routledge Press, New York, NY, 2020.

[2] J. Cuny, L. Snyder and J. Wing, "Demystifying computational thinking for non-computer scientists,". Unpublished manuscript, 2010.

[3] K-12 Computer Science Framework Steering Committee, K-12 computer science framework. ACM; 2016.

[4] J. M. Wing," Computational thinking. Communications of the ACM," Mar. 2006.

[5] J. M. Wing, "Research notebook: Computational thinking—What and why," The Link Magazine, pp. 20–23, 2011.

[6] S. Grover and R. Pea, “Computational Thinking in K–12: A Review of the State of the Field,” Educational Researcher, vol. 42, no. 1, pp. 38–43, Jan. 2013.

[7] X. Tang, Y. Yin, Q. Lin, R. Hadad, and X. Zhai, “Assessing computational thinking: A systematic review of empirical studies,” Computers & Education, vol. 148, p. 103798, Apr. 2020.

[8] J. Fraillon, J. Ainley, W. Schulz, D. Duckworth, and T. Friedman, “IEA International Computer and Information Literacy Study 2018 Assessment Framework,” p. 84, 2018.

[9] J. Lockwood and A. Mooney, “Computational Thinking in Education: Where does it Fit? A systematic literary review,” arXiv:1703.07659 [physics], Mar. 2017, Accessed: Mar. 25, 2018.

[10] M. U. Bers, L. Flannery, E. R. Kazakoff, and A. Sullivan, “Computational thinking and tinkering: Exploration of an early childhood robotics curriculum,” Computers & Education, vol. 72, pp. 145–157, Mar. 2014.

[11] A. Yadav, J. Good, J. Voogt, and P. Fisser, “Computational Thinking as an Emerging Competence Domain,” in Competence-based Vocational and Professional Education: Bridging the Worlds of Work and Education, M. Mulder, Ed. Cham: Springer International Publishing, 2017, pp. 1051–1067.

[12] M. S. Hoffer, S. Baroni, I. Fronza, and C. Pahl, “About Computational Thinking Assessment: a Proposal for Primary School First Year from a Pedagogical Perspective (invited paper),” TACKLE@EC-TEL, 2019.

[13] M. Román González, "Computational Thinking Test: Design Guidelines and Content Validation," in EDULEARN15 At: Barcelona, 2015.

[14] M. Román-González, J. Pérez-González, J. Moreno-León and G. Robles, "Can computational talent be detected? Predictive validity of the Computational Thinking Test," International Journal of Child-Computer Interaction, 2018.

[15] E. Relkin, L. De Ruiter, and M. Bers, “TechCheck: Development and Validation of an Unplugged Assessment of Computational Thinking in Early Childhood Education,” Journal of Science Education and Technology, vol. 29, Aug. 2020.

[16] M. Zapata, E. Martín, and M. Román-González, “Computational Thinking Test for Beginners: Design and Content Validation,” Apr. 30, 2020

[17] M. U. Bers, Coding as a Playground: Programming and Computational Thinking in the Early Childhood Classroom, edition 1. New York, NY: Routledge, 2018.

[18] E. Relkin, L. De Ruiter, and M. Bers, “Learning to Code and the Acquisition of Computational Thinking by Young Children,” Forthcoming. Computers & Education.

[19] S. J. Meisels, “Accountability in early childhood: No easy answers,” in School readiness and the transition to kindergarten in the era of accountability, Baltimore, MD, US: Paul H Brookes Publishing, 2007, pp. 31–47.

[20] N. Cowan, “Working Memory Maturation: Can We Get At the Essence of Cognitive Growth?,” Perspect Psychol Sci, vol. 11, no. 2, pp. 239–264, Mar. 2016.

[21] V. R. Simmering, “The development of visual working memory capacity during early childhood,” Journal of Experimental Child Psychology, vol. 111, no. 4, pp. 695–707, Apr. 2012.

[22] G. Janveau-Brennan and H. Markovits, “The development of reasoning with causal conditionals,” Developmental Psychology, vol. 35, no. 4, pp. 904–911, 1999.

[23] P. Barrouillet and J.-F. Lecas, “Mental models in conditional reasoning and working memory,” Thinking & Reasoning, vol. 5, no. 4, pp. 289–302, 1999.

[24] M. Elkin, A. Sullivan, and M. Bers, “Implementing a Robotics Curriculum in an Early Childhood Montessori Classroom,” Journal of Information Technology Education: Innovations in Practice, vol. 13, pp. 153–169, Jan. 2014.

[25] A. Strawhacker and M. U. Bers, “‘I want my robot to look for food’: Comparing Kindergartner’s programming comprehension using tangible, graphic, and hybrid user interfaces,” Int J Technol Des Educ, vol. 25, no. 3, pp. 293–319, Aug. 2015.

[26] R Core Team. R: “A language and environment for statistical computing,” R Foundation for Statistical Computing, 2019.

[27] Studio Team. “RStudio: integrated development for R,” 2018.

[28] E. Relkin, “Assessing young children’s computational thinking abilities (Master’s thesis),” Retrieved from ProQuest Dissertations and Theses database, (UMI No. 10813994), 2018.

[29] E. Relkin, and M. U. Bers, “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. Routledge, 2019.

[30] E. Marinus, Z. Powell, R. Thornton, G. McArthur, and S. Crain, “Unravelling the cognition of coding in 3-to-6-year olds: the development of an assessment tool and the relation between coding ability and cognitive compiling of syntax in natural language,” Proceedings of the 2018 ACM Conference on International Computing Education Research - ICER ’18, 133–141, 2018.

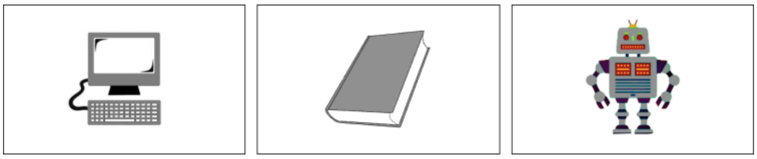
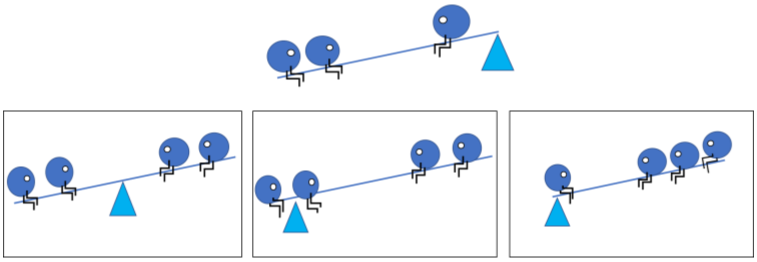
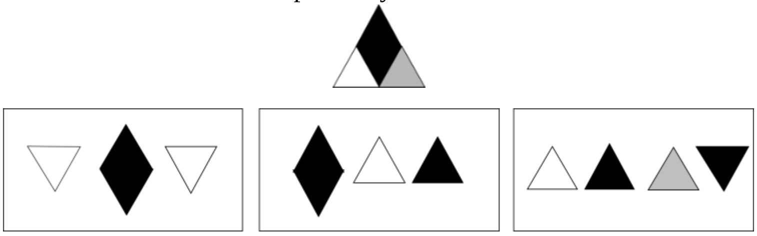
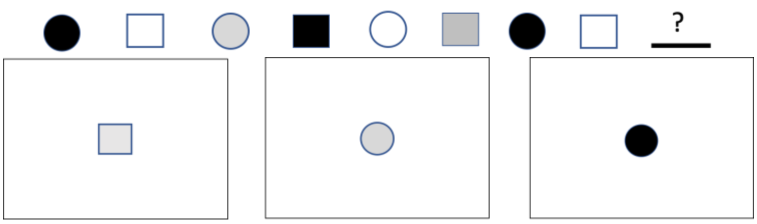
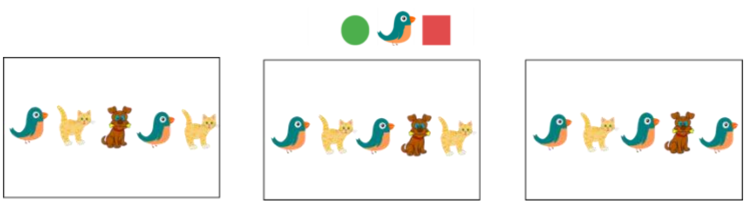
[31] C. Angeli, and N. Valanides, “Developing young children's computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy” Computers in Human Behavior, 2019.

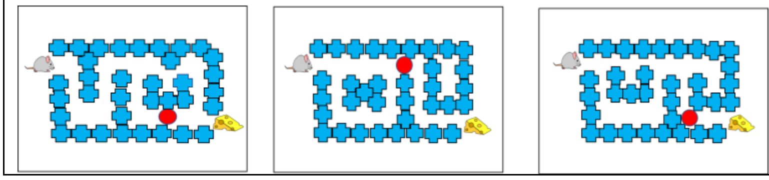
[32] M. U. Bers, “Designing digital experiences for positive youth development: From playpen to playground,” Oxford University Press, 2012.

[33] M. U. Bers, “Coding as a Playground: Programming and Computational Thinking in the Early Childhood Classroom 2nd ed,”Routledge Press, 2020.

APPENDIX

Sample *TechCheck-K* Items

Question Example	Computational Thinking Domain and Description
Which CANNOT be programmed? 	**Hardware/Software**: *Understanding the relationship between hardware and software and that technological objects are human engineered.*
This seesaw isn't going up and down. How can it be changed so it works? 	**Debugging**: *Identifying and solving a problem. Often involves perseverance, evaluation, and iterative testing.*
Which shapes can you use to make this? 	**Modularity**: *The process of breaking down larger tasks or procedures into simpler units that can be combined and reused.*
What comes next? 	**Algorithms**: *A step-by-step sequential process used to complete a task or problem.*
 A circle makes a bird and a cat. A square makes a dog and a bird. What do these shapes make? 	**Representation**: *Understanding and using the equivalency of symbols with concepts, objects, and other things.*

Mice CANNOT pass through blue walls  or red lights  Which mouse will get the cheese? 	**Control Structures:** *Recognizing and following the order and sequence in which instructions are given and events take place.*