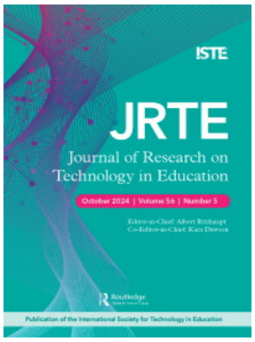

Routledge
Taylor & Francis Group

Journal of Research on Technology in Education

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/ujrt20

Validating a Creative Coding Rubric through expressive activities for elementary grades

Jessica Blake-West, Ghaida Alrawashdeh & Marina Bers

JOURNAL OF RESEARCH ON TECHNOLOGY IN EDUCATION
https://doi.org/10.1080/15391523.2024.2398502

Validating a Creative Coding Rubric through expressive activities for elementary grades

Jessica Blake-West[a], Ghaida Alrawashdeh[b] and Marina Bers[a]

[a]Boston College Lynch School of Education and Human Development, Chestnut Hill, MA, USA; [b]Department of Teaching and Learning, Oakland University School of Education and Human Services, Rochester, MI, USA

ABSTRACT

Creative coding offers the opportunity to express oneself on open-ended coding platforms through artistic expression, storytelling, and experimenting with program syntax in unconventional ways. However, research on the creative aspects of coding education and its assessment remain underexplored. This study addresses this gap by developing and validating a Creative Coding Rubric tailored for early childhood creative coding projects. Our study collected and analyzed ScratchJr projects from 1201 children in grades K-2, revealing substantial agreement among raters ($n = 15$) for the rubric. Additionally, significant positive correlations among rubric items and high reliability (α = .93) indicate its validity and consistency. This Creative Coding Rubric presents a promising tool for early childhood educational settings to evaluate creative coding skills in playful, open-ended ways.

ARTICLE HISTORY
Received 22 April 2024
Revised 26 June 2024
Accepted 12 August 2024

KEYWORDS
Creative assessment; project-based assessment; early childhood; validation; coding

Introduction

The development of computer science (CS) education is becoming increasingly important as the technological world evolves (Montuori et al., 2024; Moreno León et al., 2016; Mouza et al., 2022). In the past decade, there has been a notable emphasis on bringing coding education and digital literacy skills to younger grades (Tang et al., 2020; Wakil et al., 2019). CS encompasses many different domains of thinking such as problem solving, computational thinking (CT), design, and expression, all of which can be accessed through developmentally appropriate practices such as open-ended play and tinkering (Girshin et al., 2023; Kafai & Burke, 2014).

Creative coding projects engage children as producers and not merely consumers of technology. Coding can serve as an expressive medium that taps into emotional and social domains, fostering a deeper understanding and appreciation of technology while developing CT (Bers, 2021). Encouraging the creation of personally meaningful projects provides children with opportunities to explore their creativity, develop problem solving skills, and express themselves through computational innovations.

In recent years, there has been a push to emphasize creative and expressive aspects of CS through the use of creative coding platforms and project-based assignments, alongside more traditional CT focused, task-based assignments (Fragapane & Standl, 2021; Hendrickson et al., 2021; Sharmin, 2021). This emphasis is particularly pronounced in early childhood education, where hands-on, explorative activities align with young children's developmental needs (Hirsh-Pasek et al., 2020). Until recently, however, there has been a scarcity of resources designed to meet

CONTACT Jessica Blake-West jess.blake-west@bc.edu Boston College Lynch School of Education and Human Development, Chestnut Hill, MA, USA

Supplemental data for this article can be accessed online at https://doi.org/10.1080/15391523.2024.2398502.

the needs of young learners in creative CS education. To address this need, tools such as ScratchJr and the Foos from Codespark.org and curricula like the Coding as Another Language Curriculum (CAL) have been developed (Bers et al., 2023).

There have also been promising developments in research on creative coding assessments for young children. Task-based assessments and project-based assessments have been developed and tested, both of which demonstrated good psychometric properties (Guttman's $\lambda 6 = .94$; Krippendorf's $\alpha = .95$; respectively de Ruiter & Bers, 2022; Unahalekhaka & Bers, 2022). Despite these developments, research on early childhood creative coding projects is still very limited. Building upon previous work of Unahalekhaka & Bers (2022), offered in the Appendix in ScratchJr Project Rubric section, we sought to further develop a creative coding rubric which recognizes the multiple ways creativity and CT can be expressed through ScratchJr coding projects. ScratchJr is used by over 50 million users around the world, and thus by developing a rubric with adaptations to this free and widely used platform, we offer an applicable assessment tool for teachers and researchers worldwide. Through this development process, we redesign items to measure creative coding as a single underlying construct, rather than two distinct constructs– project design and coding concepts.

Our aim is to expand the breadth and variety of assessment tools available which encompass creative expression through coding. Our research questions are as follows:

RQ1. Is a one-factor structure a valid framework for capturing variations in children's coding choices and expressions during self-guided activities within an open-ended programming environment?

RQ2. Does the proposed rubric, based on a single underlying construct, demonstrate reliability and validity in assessing children's coding activities, including their choices and expressions, over time?

Through this investigation, we aim to contribute to the ongoing efforts to develop robust assessment tools for evaluating elementary children's coding skills and creative expression, advancing our understanding of how coding can foster CT and creativity in early childhood education.

Literature review

Computer science in early childhood

Technological fluency, as acquired through CS educational initiatives, is crucial for living in the twenty-first century (Nouri et al., 2020; Unahalekhaka & Bers, 2022). There are many tools and curricula developed for CS in elementary school such as The Creative Computing Curriculum for Scratch and the Adventures in Alice Programming.[1] However, these tools focus on ages eight and up, with no support for early childhood. In the past decade there has been a push for coding education in early childhood settings (Barron et al., 2011; The White House, n.d.; Weissman et al., 2023). This movement is gaining momentum, fueled by policy initiatives like "CS for All" in the U.S. and the inclusion of CS as a mandatory subject in countries like Australia and England (Brown et al., 2014; Falkner et al., 2014; Mouza et al., 2022). To nurture this vision, new standards from the Computer Science Teachers Association and the International Society for Technology in Education champion a paradigm shift toward fostering foundational CS knowledge and abilities amongst K-12 students (Mouza et al., 2022).

This is particularly crucial in light of research highlighting that younger children are eager to gain technological fluency given the changing digital landscape (Popat & Starkey, 2019; Sun & Zhou, 2023). Further, studies have shown that coding education enables children to organize and process information in more effective and creative ways (Kim & Jeong, 2023). Crucially, introducing coding as early as preschool age capitalizes on this developmental window, as young children are naturally transitioning from concrete to abstract thought processes, making them highly receptive to the foundational building blocks of CT (Bati, 2022).

In order to teach CS to young children in developmentally appropriate ways, numerous tools have been created for young children including Kodable, LightbotJr, Code.org, Codeable Crafts, The Foos, Cargo-Bot, Move the Turtle, and ScratchJr (See Coding Platform Comparison section in the Appendix). Many of these tools utilize visual, block-based coding and playful animations and/or games, to introduce concepts such as sequencing, loops, and conditionals to young children. Among these tools, ScratchJr[2] stands out as one of the most popular and freely accessible platforms for young children, and has been shown to effectively familiarize young children with CS concepts (Papadakis et al., 2016; Yang & Bers, 2023).

In addition, the development of coding curricula and learning resources tailored for early childhood CS education has grown to support the use of these tools. Notable examples include the Coding as Another Language (CAL) curriculum for both ScratchJr (Yang et al., 2023) and KIBO robotics (Ben Ari et al., 2023). Despite this promising development, there remains relatively little research regarding the intervention and evaluation of early childhood coding education (Chou, 2020; Papavlasopoulou et al., 2020; Relkin et al., 2020; Sun et al., 2023).

In a scoping review on early childhood computer science education, Su et al. (2023) found that coding curriculum design, implementation and effectiveness in early childhood education settings are understudied. Su et al. (2023) also noted that only the curriculum developed by Monteiro et al. (2021) integrated a greater emphasis on open-ended creative coding practices.

Creative coding

We conceptualize creative coding as the innovative application of CS skills to express oneself (Resnick, 2017). This practice encompasses a diverse array of approaches, including the utilization of customization and artistic features, crafting narratives or games through code, and exploring unconventional syntax techniques. These creative avenues, alongside others, can be promoted through employing play-based pedagogy, tools, curricula, and teaching strategies.

Coding provides children with a new language which promotes new modes of thinking, communication and idea expression (Girshin et al., 2023; Kafai & Burke, 2014). Coding has been shown to increase creativity, critical thinking, and problem solving (Canbeldek & Isikoglu, 2023; Chun & Park, 2020; Grover & Pea, 2013; Hu, 2023; Popat & Starkey, 2019; Yang et al., 2023). Furthermore, it has been shown that the creation of customized projects promotes both coding skills and creativity (Sun et al., 2023; Wang et al., 2023). Creative coding, promoted through open-ended prompts, customizable environments, and a wide variety of syntax and functionality, allows for both creative and computational skills to emerge. Booton et al. (2023) demonstrated a gap in creative educational digital tools for young children. ScratchJr, among other creative coding apps like The Foos, address this need through integrating multiple customizable design features and creative functionalities.

Project-based learning stands out as a primary method for fostering child-centered creative coding because it allows children to explore coding and CT concepts in hands-on, meaningful ways. Open-ended projects are shown to promote agency in learners, promote creative engagement as well as facilitate deep learning and knowledge application (Barron & Darling-Hammond, 2008; Grover et al., 2018). Notably, in one study, researchers explored the impact of using storytelling as a framework for ScratchJr project-based creative coding in early childhood education (Yang et al., 2023). In this present study, we adopt a similar approach by integrating storytelling through ScratchJr as a framework for project-based creative coding in K-2.

Assessments

Across all ages, assessments serve to inform and scale up education while providing feedback to learners to prevent the widening of learning gaps (Basu, 2019; Pyle et al., 2022). The recent focus on educational accountability, particularly evident in the United States, has led to increased presence of assessment in younger grades (Broadhead, 2006; Goldstein et al., 2017; Gullo &

Hughes, 2011; Pyle et al., 2020, 2022). This has proven difficult for many early childhood educators trying to maintain play-based pedagogies (DeLuca et al., 2020a; Gullo & Hughes, 2011; Jeynes, 2006; Lynch, 2015). Performance-based assessments with flexible, open-ended tasks that account for play-based pedagogies offer a promising solution for evaluating learning in early grades (Basu et al., 2018; Bortz et al., 2020; Sherman & Martin, 2015; Weintrop et al., 2014).

However, a crucial gap exists in research on assessment for younger students, particularly in early childhood and elementary settings (Pyle et al., 2020). Research specifically examining how assessments can be integrated effectively with play-based learning methods is even more limited. This necessitates reimagining assessment practices to fit pedagogical approaches like play-based learning in early childhood education, and requires increased support from policymakers to bridge this research gap and implement these changes effectively (DeLuca et al., 2020b).

Coding assessments in early childhood

Research in early childhood coding education has seen a surge in the development of evaluation methods (Macrides et al., 2022). However, these assessments often fall short, primarily relying on observational or task-based approaches rather than leveraging the potential of artifacts (Macrides et al., 2022; Sun et al., 2023). Task-based approaches are particularly prevalent in coding evaluation (See Angeli & Valanides, 2020; de Ruiter & Bers, 2022; Di Lieto et al., 2017; Pila et al., 2019; Sung et al., 2017). Furthermore, Cutumisu et al. (2019) and Tselegkaridis & Sapounidis (2022) identified a widespread reliance on multiple-choice questions for assessing CT skills. Despite these tools, many researchers still identify a need for validated tools for both early childhood coding and CT skills (Angeli & Valanides, 2020; Ioannou & Makridou, 2018; Macrides et al., 2022). This gap is further amplified by a lack of research on assessment methods specifically designed for open-ended projects.

Creative coding assessments

Open-ended projects offer an effective, authentic, and commonly used approach to assess CS skills, particularly in younger learners (Basu, 2019; Grover et al., 2018; Unahalekhaka & Bers, 2022). While rubrics remain the dominant method for assessing project-based learning (Al Kandari & Al Qattan, 2020; Basu, 2019; Bortz et al., 2020; Catete et al., 2016; Cutumisu et al., 2019; Grover et al., 2018; Moreno León et al., 2016), most existing rubrics exclude both early childhood education and creative coding (Basu, 2019; Denner et al., 2012; Grover, 2020; Moreno-León & Robles, 2015; Unahalekhaka & Bers, 2022; Wilson et al., 2013).

Recognizing this need, Unahalekhaka & Bers (2022) pioneered the ScratchJr Project Rubric, evaluating coding skills and project design across 13 categories. Validated with 228 ScratchJr projects, the rubric demonstrated validity and reliability. Building upon this work, we present the Creative Coding Rubric. This new rubric addresses three key concerns identified from valuable user feedback to increase the rubric's ease of use for educators and researchers alike. We shifted the focus from isolated coding concepts and project design elements toward a holistic evaluation of creative coding skills. This involved consolidating and adjusting measures within the rubric to reflect this broader aim. In doing so, we redefined criteria to capture the creative coding thought processes, moving beyond interface-specific skills and enabling application of the rubric across diverse programming environments. Through these advancements, we aim to establish an objective tool that allows for the examination of children's creative coding skills through the lens of project-based learning.

Methods

ScratchJr

While the Creative Coding Rubric is versatile and can be adapted to various environments, for the validation data used in this study, we focused on projects created within ScratchJr.

This block-based programming language, specifically designed for young minds, empowers children to grasp commands through intuitive symbols instead of text-heavy instructions (Flannery et al., 2013). Unlike their text-based counterparts, block-based languages like ScratchJr eliminate the need for complex syntax, making them more accessible for young learners (Grover, 2020). ScratchJr offers six coding block categories: Triggering or "Start", Motion, Looks, Sounds, Control, and End, each with its own function and color-coded for easy identification as can be seen in Figure 1. These features allow children to creatively combine blocks to design animated stories with custom backgrounds and characters.

Figure 1 shows ScratchJr interface. In the programming area users can program character interaction, motion, sound, appearance changes, and setting changes. On the center of the figure (the stage), all program output is displayed as animations. Characters can be customized (see Cat 5 in Figure) and created from scratch (See Character 1 and 2 in Figure), and all can be programmed with different codes. Projects can also contain multiple pages with customizable backgrounds. In Figure 1, a hand-drawn character is being programmed to change size, repeat a set of movements, play a recorded sound, and then go to a new page.

Rubric development

Incorporating feedback from both users and experts, the Creative Coding Rubric expands upon the framework established by the previous ScratchJr Project Rubric (Unahalekhaka & Bers, 2022), offering a streamlined and comprehensive approach to assess creative coding.

While the ScratchJr Project Rubric covers various creative coding concepts, its previous organization tends to focus on interface-specific aspects. Consequently, many categories in the original rubric assess multiple coding concepts, and several concepts extend across multiple categories. To address this, we first identified six main creative coding concepts and mapped those concepts to the previous rubric. Using these overarching creative coding concepts as a guide, we realigned the concepts to the ScratchJr programming environment, resulting in the Creative Coding Rubric shown in Figure 2.

Figure 1. ScratchJr interface showing visual and programmable creative affordances.

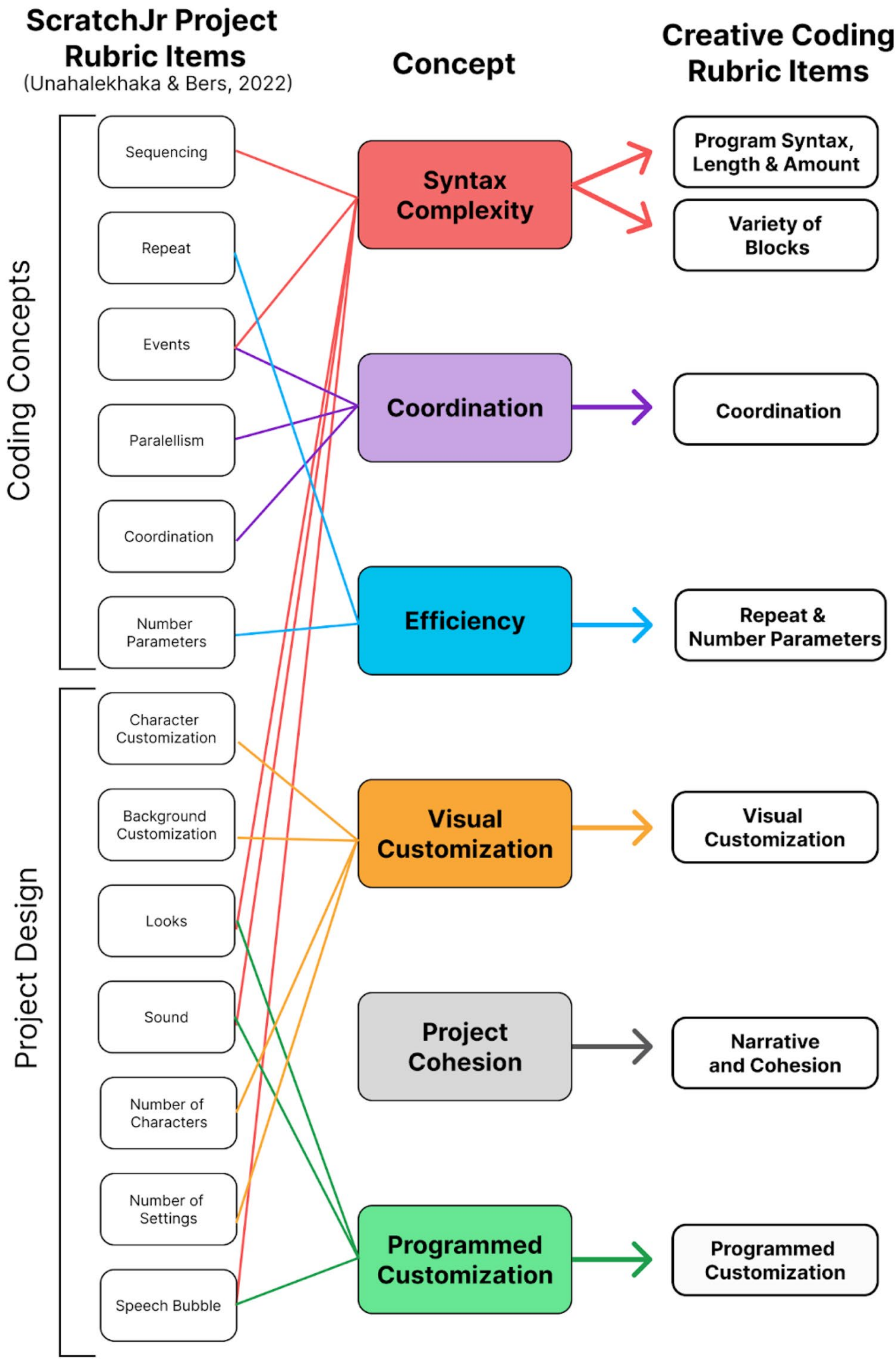

Figure 2. Mapping the creative coding concepts onto the ScratchJr project rubric and creative coding rubric.

The first creative coding concept we identified was *Syntax Complexity*, assessed through sequencing, events, and use of "look", "sound", and "speech bubble" blocks in the original ScratchJr Project Rubric. In our first iteration, we attempted to consolidate these categories to focus on two areas of syntax complexity: the quantity and diversity of functions. To measure the quantity of functions, we introduced a category titled 'program length, syntax, and amount,' which encompasses similar metrics to the previous 'sequencing' category. This category assesses the length and quantity of code sequences while also considering code structure and complexity, previously evaluated under 'events.' In assessing the variety of functions, we introduced a category named 'block variety,' which replaces interface-specific categories such as 'look,' 'sound,' 'speech bubble,' as well as 'events.'

However, upon piloting, we concluded that the two approaches to evaluating syntax complexity were essentially assessing the same skills. This observation was reinforced by the high correlation between the categories observed during piloting. Consequently, in the final version of the Creative Coding Rubric, we combined these two aspects to provide a comprehensive assessment of syntax complexity.

The subsequent concept we identified was *Coordination*, previously assessed across three categories: events (focusing on the use of the 'Start on bump' and 'Go to page' blocks in ScratchJr), parallelism, and coordination (specifically examining the 'Set Speed,' 'Wait,' and 'Stop' blocks in ScratchJr). Upon synthesizing these categories, we recognized that each measured various aspects of how children acquire coordination skills through coding. As a result, we merged these skills into a single category in our iteration.

Another coding concept we identified was *Efficiency*. Efficiency in ScratchJr can be understood and practiced through the use of number parameters instead of sequential identical blocks, as well as through the implementation of repeat loops to iterate sequences of blocks. In the original rubric, number parameters and repeat loops were assessed as separate categories. However, it was observed through testing that this approach resulted in inconsistencies in scoring across categories, as parameters are typically an introductory step toward understanding the more advanced application of efficiency with repeat loops. Therefore, in our iteration, we combined these two measures into a single item titled 'Repeat and Number Parameters,' which delineates the natural skill progression from using number parameters to employing loops, and ultimately to integrating both techniques including nested loops.

The next creative coding concept we focused on was *Visual Customization*. The original rubric assessed this skill across four categories: character customization, background customization, number of characters, and number of settings. Although each category demonstrates a distinct aspect of visual customization within the ScratchJr environment, the underlying skill being measured remains consistent across all categories. Therefore, we combined these categories to represent a single item.

Another creative coding concept we identified was *Programmed Customization,* which examines children's use of coding functions and commands in creative and expressive ways to enhance their project. In the original rubric, separate categories existed for the "Look", "Sound" and "Speech Bubble" blocks. However, as mentioned in the discussion on block variety, these categories are highly interface-specific, whereas the actual skill being assessed is the number of functions used to customize the project further. Therefore, in our rubric iteration, we consolidated each of the ScratchJr-specific functions into one item representing the general practice of programmed customization.

Moreover, in this category, we adjusted the original grading criteria to account for children's customizations that previously went unrecognized. In the original rubric, children received no credit for using blocks unattached to a sequence or start block. Consequently, many children who used voice recordings separately from the rest of their code, for instance, were not given any credit. Recognizing the nature of the ScratchJr platform, specifically that blocks can be played without being attached to a sequence, we modified this rule only when assessing programmed customization. For categories such as program syntax or coordination, the use of full syntax remained a requirement. However, when solely evaluating a child's customization abilities with blocks, we avoided imposing additional penalties for syntax errors.

Lastly, we identified a creative coding concept that was not previously measured in the ScratchJr Project Rubric: *Project Cohesion*. Evaluating this aspect is admittedly challenging, as Unahalekhaka & Bers (2022) discuss, since conveying cohesion and narrative often depends on the coder's intentions which are not always apparent from the project alone and may require an interview or conversation. Nevertheless, leveraging the expressive tools available on a creative coding platform, such as ScratchJr, and by examining connections between codes, we believed that assessing cohesion could be possible.

To address this, we devised an item called "Narrative and Cohesion," which probes objective measures of a coding project, assessing how its elements adhere to one another. This item holds particular significance in evaluating creative coding, given the notion that coding can serve as a language for expression (Bers et al., 2023). Given the expressive nature of conveying narrative and cohesion, and the affordances offered by many of the programmed customization functions for natural language, we anticipate a potential correlation between the two items. For instance, a child may receive a higher narrative score if they use a voice recorder block to narrate their story, thereby also enhancing their programmed customization score.

In total, the Creative Coding Rubric comprises six scoring categories, with two subcategories for syntax complexity, offering five possible scores each: 0, 1, 2, 3, 4. Each box in the rubric delineates the criteria necessary for a project to receive that specific score. In response to user feedback, we included a full page of ScratchJr interface-specific tips and coding term definitions. Multiple iterations resulting from the revision process underwent thorough review and testing by experts in the field, and adjustments were made based on their feedback. To access the Creative Coding Rubric, please refer to the Appendix.

Participants and sampling

The participant pool consists of 1,201 children from 32 schools across five states. Approximately 88% of the children were drawn from 27 schools located in New England. Among the sampled children, 26% were in kindergarten, 38% were in first grade, and 36% were in second grade. In Year 1 of data collection, Grade 1 had the highest participation with 63%, followed by Grade 2 with 33%. In Year 2, Grade 2 saw the highest participation with 44%, followed by Grade 1 with 28%. In Year 3, Grade 1 had the highest participation again with 41%, followed by Grade 2 with 30%.

About 14% of the children qualified for an Individualized Educational Program (IEP), and 11% were enrolled in a Language Enrichment Program (LEP). In terms of race, the majority of children identified as White (68%), followed by Hispanic (17%) and Black or African American (5%). Asian and Mixed or Other racial groups each constituted a smaller percentage, at 3% and 6% respectively. Half of the children identified as female, with the majority (74%) classified as high socioeconomic status (SES), while 22% indicated low SES.

All the sampled children were in classrooms that taught the CAL curriculum using ScratchJr. Throughout the curriculum, each child was requested to create three open-ended, creative projects in ScratchJr. The first two projects were requested in the first two quarters of the curriculum: project 1 around Lesson 6 and project 2 around Lesson 12 (specific project lessons differ across the K-2 curricula). Then, for all grades, the final project spans lessons 19-24. We will be referring to these projects in our analysis as time points 1, 2, and 3, respectively. The data collection lasted for 3 years (from SY 2020-2021 to SY 2022-2023), with different participants each year. For a sample of these projects, please refer to the Appendix.

At Time Point 1, a total of 809 projects were submitted, accounting for 67% of the total submissions. Time Point 2 saw 328 (27%) submissions, while at Time Point 3, there were 64 submissions (5%). On average, children were proficient, scoring 13, with a minimum of 0 and a maximum of 28.

All projects were de-identified, using a random ID number. Once de-identified, a total of 16 research assistants (referred to as raters hereafter) were hired to complete the grading of each project. However, one rater was removed from the data set due to inconsistent entries, and all projects from that rater were re-graded by another trained rater, resulting in a sample of 15 raters. Table 1 presents the number of ratings (n) and the percentage (%) of total ratings contributed by each rater, alongside demographic information. The distribution of ratings among the raters varied from 9 to 289, with Rater 2 contributing the highest percentage of ratings (24%), followed by Rater 9 (13%).

Analysis

Our analysis encompassed an examination of Inter-Rater Reliability (IRR) and rubric reliability. For IRR, we used Fleiss' Kappa, while for evaluating the reliability of the rubric, we used statistical techniques such as Pearson correlation and Cronbach's alpha. To explore the underlying structure of the rubric, we conducted Exploratory Factor Analysis (EFA) and Confirmatory Factor Analysis (CFA). We extended the analysis to assess reliability across multiple time points using hierarchical linear modeling (HLM) to gauge the consistency of rubric scores over time.

Rater agreement

In this study, all raters underwent an hour-long video training session, during which example scenarios for each aspect of the rubric were provided. Subsequently, each rater participated in a reliability test before being cleared to begin grading. In this test, raters were presented with the same five sample projects and tasked with grading each one. In order to be cleared to grade, raters needed to grade all projects within 1 point of the correct grade. All raters were offered optional check-ins with the research team to review the rubric as needed before taking the reliability test. Initially, all raters independently assessed each project, resulting in a dataset of initial ratings. Following this, feedback was provided based on the aggregated results, and raters were given the opportunity to revise their ratings for each project. The ratings for each project were then categorized into the following groups: (0-1 points) Budding, (2-7) Developing, (8-14) Proficient, (15-21) Advanced, and (22-28) Distinguished. To quantify the consistency and agreement among raters' ratings, Fleiss' Kappa was calculated separately for both the initial and revised rounds of ratings using the IRR package in R (Gamer et al., 2019).

Rubric reliability

To validate the reliability of the rubric, we started with a Pearson correlation analysis to evaluate the inter-item relationships. Cohen's standard was used to evaluate the strength of the relationship (Cohen, 1988).[3] The result was examined using the Holm correction to adjust

Table 1. Frequency table for the number of ratings (n) by each rater alongside demographic information.

Rater	n	%	Age	Gender	Race
Rater 1	80	7	20	Man	White
Rater 2	289	24	20	Woman	White
Rater 3	87	7	19	Woman	Hispanic or Latino
Rater 4	31	3	19	Woman	Prefer not to respond
Rater 5	82	7	21	Woman	Asian
Rater 6	88	7	20	Woman	White
Rater 7	38	3	20	Woman	White
Rater 8	14	1	19	Prefer not to respond	Prefer not to respond
Rater 9	152	13	20	Woman	Asian
Rater 10	85	7	19	Woman	Asian
Rater 11	56	5	19	Woman	White
Rater 12	9	1	20	Woman	Asian
Rater 13	62	5	20	Woman	Asian
Rater 14	88	7	19	Man	Asian
Rater 15	40	3	20	Prefer not to respond	Prefer not to respond

for multiple comparisons based on an alpha value of .05. This was then followed by calculating a Cronbach alpha coefficient for the scores for the seven items. The Cronbach's alpha coefficient was evaluated using the guidelines suggested by George & Mallery (2019).[4] While the Pearson correlation analysis does not directly measure internal consistency like Cronbach's alpha does, it was a complementary analysis to help validate the reliability of the rubric.

Factor analysis

To investigate whether the rubric comprised distinct categories—coding concepts and design concepts—or a unified construct, we started with running an EFA on the seven variables. For this analysis, we generated a composite score by summing the scores for each item across all time points. At each time point, children were presented with an open-ended question. However, in the second and third instances, children were required to demonstrate their proficiency in incorporating advanced coding skills. Aggregating the scores across time allowed us to evaluate the overall performance of children as they demonstrated their ability to apply increasingly sophisticated coding techniques.

For the EFA, we used both the Kaiser criterion and parallel analysis to determine the number of factors to retain with Promax rotation. All variables exhibited correlation coefficients exceeding .30, suggesting suitability for factor analysis (Tabachnick & Fidell, 2019). The determinant value of the correlation matrix, 0.0010, indicated the absence of multicollinearity in the data (Field, 2017). Following Tabachnick and Fidell's recommendation (2019), the minimum threshold for identifying significant factor loadings was set at .32. With a participant-to-item ratio of approximately 171 to 1—based on a sample size of 1201 and seven variables included—the analysis suggested ample data for producing reliable results (Costello & Osborne, 2005).

This analysis was then followed by conducting a CFA model to determine whether the one factor model (Creative Coding) adequately described the data. We used Maximum likelihood (ML) estimation to determine the standard errors for the parameter estimates. None of the variables exhibited an $R^2 > .90$, and the determinant value for the correlation matrix was 0.008, indicating no multicollinearity after combining items 1 and 2 (Field, 2017). The participant to item ratio for this analysis was approximately 92 to 1, where sample size was 1201 and the number of variables included was 13. This sample size was sufficient to produce reliable results (Kline, 2015). In the CFA model, items 1 (Program Syntax, Length, and Amount) and 2 (Variety of Blocks) were combined while items 6 (Narrative and Cohesion) and 7 (Programmed Customization) were allowed to covary as discussed earlier in the Rubric Development section.

Rubric reliability across occasions

We also examined the reliability of the rubric across time points. Specifically, we conducted a series of HLM models to assess whether the total project scores (sum of all items; *Score*) could be significantly predicted by *Time* (controlling for demographic variables) while considering the nested structure of the data. Assessing the assumptions, a total of 9 outliers were detected which were addressed using the Winsorization method.[5] The Intercept-Only Mode, served as a null model, regressing Score solely on the intercept, with random intercepts for each *Child* to capture individual variability. Subsequent models expanded upon this framework. Time-Only Model introduced Time as the sole predictor, representing the effect of time on the rubric scores, while still incorporating random intercepts for each child. The Expanded Model expanded the scope by including additional demographic predictors: *Gender*, *SES*, *LEP*, *Grade*, *IEP*, and *Race*. Random intercepts for each child remained to address individual variability. However, when attempting to incorporate a random slope for Time to allow its effect to vary across children, the model did not converge, suggesting challenges in model estimation.[6]

Throughout the analysis, ML estimation was used. To evaluate the overall fit of the model, we conducted a likelihood ratio test by comparing the Expanded Model, with the Intercept-Only Model. For the assessment of fixed effects, *p*-values were estimated based on a standard normal distribution. This analysis was conducted based on the imputed data using the lme4 package (version 3.1-164; Bates et al., 2023).

Results

Rater agreement

We sought to evaluate the IRR of ratings provided by the 15 raters for the set of five projects using Fleiss' Kappa given that the final scores were one of the five categories described earlier (see the Creative Coding Rubric section in the Appendix). The IRR analysis for the initial round of ratings yielded a Fleiss' Kappa coefficient of 0.5, indicating a moderate agreement[7] among the raters beyond what would be expected by chance ($z = 16.3$, $p < 0.001$). For the second round of ratings, Fleiss' Kappa was again computed for the same group of 15 raters evaluating the same five projects. The analysis revealed an improved Fleiss' Kappa coefficient of 0.7, indicating substantial agreement among the raters beyond chance ($z = 21.1$, $p < 0.001$). The agreement was particularly strong for the higher proficiency categories "Proficient" (Kappa = 0.8, $z = 18.2$, $p < 0.001$) and "Distinguished" (Kappa = 0.8, $z = 17.5$, $p < 0.001$).

Rubric reliability

According to the result of the Pearson correlation analysis, significant positive correlations were observed among all rubric items. All the correlations were above .50 indicating a large effect size (Cohen, 1988), except for the correlations between item 4 and items 5, 6, 7 which had moderate effect size. The highest correlation was found between items 1 and 2 as anticipated ($p < .001$, 95.00% CI = [.93, .94), further supporting our decision to merge them into a single item during our factor analysis. Table 2 presents Pearson correlation matrix among the items and Table 3 presents the results of the correlation analysis. The seven items had a Cronbach's alpha coefficient of .93 (95% CI [.93, .93]), indicating excellent reliability.

Factor analysis

To investigate the underlying structure of the rubric, we started with an EFA applying both the Kaiser criterion and Parallel analysis for electing how many factors to retain. Evaluation of the scree plots (see Figure 3, panels A and B) revealed that one factor exhibited an eigenvalue exceeding one or greater eigenvalue than its randomly generated counterpart. As a result, we proceeded with a single factor for the factor analysis.

The one-factor model, representing creative coding, accounted for approximately 69% of the variance with an eigenvalue of 4.83. However, a Chi-square goodness-of-fit test indicated that the one-factor model did not adequately depict the data, $\chi^2(14) = 295.97$, $p < .001$. It is important to note that this test is sensitive to sample size. The factor analysis summary is presented in Table 4. Notably, all loadings in our analysis exceeded 0.65. Lee & Comrey, (2013) suggest that factor loadings between .63 and .71 are considered very good. Additionally, there were no

Table 2. Pearson correlation matrix among the rubric items.

Variable	1	2	3	4	5	6	7
1. Item 1	–						
2. Item 2	.93*	–					
3. Item 3	.90*	.88*	–				
4. Item 4	.62*	.68*	.58*	–			
5. Item 5	.74*	.73*	.69*	.44*	–		
6. Item 6	.67*	.65*	.66*	.42*	.57*	–	
7. Item 7	.74*	.77*	.74*	.47*	.59*	.68*	–

Table 3. Pearson correlation results among the seven items ($n = 1201$).

Combination	r	95.00% CI	p
Item 1 - Item 2	.93	[.93, .94]	< .001
Item 1 - Item 3	.90	[.88, .91]	< .001
Item 1 - Item 4	.62	[.59, .66]	< .001
Item 1 - Item 5	.74	[.72, .77]	< .001
Item 1 - Item 6	.67	[.63, .70]	< .001
Item 1 - Item 7	.74	[.71, .76]	< .001
Item 2 - Item 3	.88	[.87, .90]	< .001
Item 2 - Item 4	.68	[.65, .71]	< .001
Item 2 - Item 5	.73	[.70, .75]	< .001
Item 2 - Item 6	.65	[.62, .69]	< .001
Item 2 - Item 7	.77	[.75, .79]	< .001
Item 3 - Item 4	.58	[.54, .62]	< .001
Item 3 - Item 5	.69	[.66, .72]	< .001
Item 3 - Item 6	.66	[.63, .69]	< .001
Item 3 - Item 7	.74	[.71, .76]	< .001
Item 4 - Item 5	.44	[.40, .49]	< .001
Item 4 - Item 6	.42	[.37, .46]	< .001
Item 4 - Item 7	.47	[.42, .51]	< .001
Item 5 - Item 6	.57	[.53, .61]	< .001
Item 5 - Item 7	.59	[.55, .62]	< .001
Item 6 - Item 7	.68	[.64, .71]	< .001

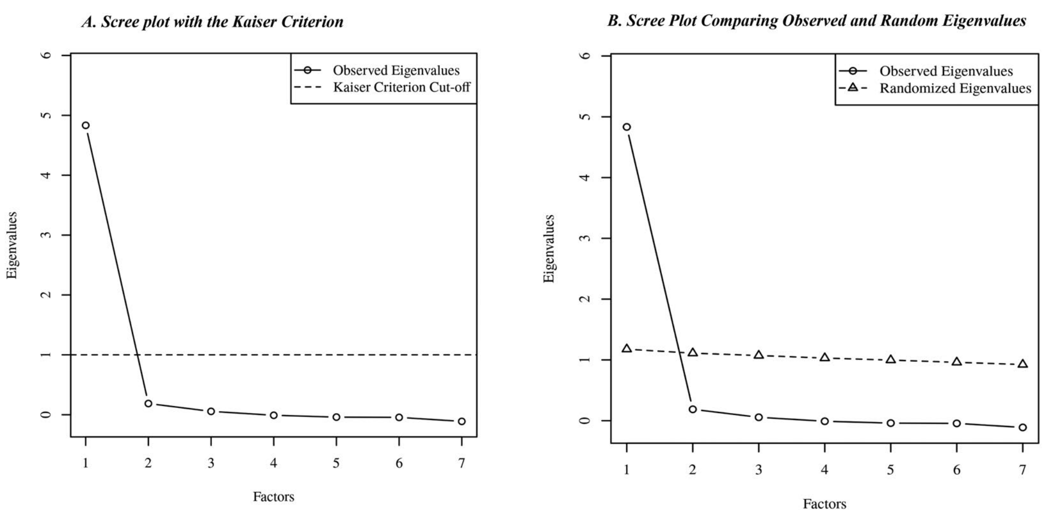

Figure 3. Scree Plots for EFA Analysis with Panel A Showing Results for the Kaiser Criterion and Panel B Comparing Observed and Random Eigenvalues for Parallel Analysis.

Table 4. Factor loadings from exploratory factor analysis.

Variable	Factor Loading	Communality
Sum of Item 1	0.96	0.93
Sum of Item 2	0.97	0.93
Sum of Item 3	0.92	0.85
Sum of Item 4	0.66	0.44
Sum of Item 5	0.76	0.57
Sum of Item 6	0.70	0.49
Sum of Item 7	0.79	0.62

Note: $\chi^2(14) = 295.97$, $p < .001$; Eigenvalue = 4.83; % of variance = 69; $\alpha = .93$.

variables with low communalities (< .40), indicating that the factor structure describes the data well (Costello & Osborne, 2005). Further, the absence of variables with cross-loadings suggests a simple and easily interpretable factor structure. Each factor exhibited at least three significant loadings (> .32), indicative of a strong and robust factor (Costello & Osborne, 2005). The items corresponding to this factor demonstrated a Cronbach's alpha coefficient of .93, reflecting excellent reliability (see Table 4).

The results of the CFA model are presented in Table 5. All the fit indices indicate that the model is a good fit for the data, except for the Chi-square goodness of fit test, $\chi^2(8) = 83.78$, $p < .001$, which, as mentioned earlier, is sensitive to sample size (Hooper et al., 2008). There were no observed variables with R^2 values ≤ .20 (Hooper et al., 2008). The R^2 values, along with the error variances for each observed variable are presented in Table 6.

Rubric reliability across occasions

Next, we examined the reliability of the rubric scores across multiple time points, while considering various demographic variables. Based on the HLM analysis, the likelihood ratio test for the HLM model was significant based on an alpha of .05, ($\chi^2(13) = 186.87$, $p < .001$), indicating that the Expanded Model provides a significantly better fit to the data compared to the Intercept Only Model. This indicates that adding the time variable to the model accounted for the variation in children scores better than the mean. Subsequently, the Time-Only Model introduced the Time variable, revealing a significant positive association with scores as summarized in Table 7 ($\beta = 0.719$, $p < .01$). Expanding the model further to include demographic predictors yielded the Expanded Model. Notably, while controlling for these variables, the Time-Only Model remained consistent, with a slightly reduced coefficient ($\beta = 0.661$, $p < .01$). Across all models, certain demographic factors showed significant associations with scores, including Grade, Asian race, and Not Reported SES. Additionally, the Expanded Model exhibited an improved marginal and conditional R-squared compared to the Intercept-Only Model, indicating enhanced

Table 5. Unstandardized loadings (standard errors), standardized loadings, and significance levels for each parameter in the CFA model ($N = 1201$).

Parameter Estimate	Unstandardized	Standardized	*p*
Loadings			
Creative → Sum of Items 1&2	1.00(0.00)	0.99	–
Creative → Sum of Item 3	0.32(0.005)	0.92	< .001
Creative → Sum of Item 5	0.36(0.010)	0.76	< .001
Creative → Sum of Item 4	0.13(0.004)	0.66	< .001
Creative → Sum of Item 6	0.29(0.009)	0.69	< .001
Creative → Sum of Item 7	0.37(0.009)	0.78	< .001
Covariances			
Covariance for Sum of Item 6 and Sum of Item 7	0.82(0.08)	0.31	< .001
Errors			
Error in Sum of Item 3	0.56(0.03)	0.16	< .001
Error in Sum of Item 5	2.83(0.12)	0.43	< .001
Error in Sum of Item 4	0.63(0.03)	0.56	< .001
Error in Sum of Item 6	2.74(0.11)	0.53	< .001
Error in Sum of Item 7	2.59(0.11)	0.39	< .001
Error in Sum of Items 1&2	0.85(0.19)	0.03	< .001
Error in Creative	28.77(1.22)	1.00	< .001

Note. $\chi^2(8) = 83.78$, $p < .001$; NFI & CFI = 0.99; TLI = 0.98; RMSEA (90% CI = [0.07, 0.11]) = 0.09; SRMR = 0.02; – indicates the statistic was not calculated due to parameter constraint.

Table 6. Estimated error variances and R^2 values for each indicator variable - latent variable relationship in the CFA model.

Endogenous Variable	Standard Error	R^2
Sum of Items 1&2	0.85	.97
Sum of Item 3	0.56	.84
Sum of Item 5	2.83	.57
Sum of Item 4	0.63	.44
Sum of Item 6	2.74	.47
Sum of Item 7	2.59	.61

Table 7. Hierarchical linear modeling results for the creative projects with robust standard errors within parentheses ($n = 1201$ children, 2084 observations).

Variable	Intercept-Only Model	Time-Only Model	Expanded Model
(Intercept)	13.866***	12.906***	10.496***
	(0.140)	(0.347)	(0.410)
Time Point		0.719**	0.661**
		(0.241)	(0.235)
Gender (Female)			0.025
			(0.269)
SES (Not Reported)			−1.195*
			(0.605)
Low SES (Yes)			−0.056
			(0.350)
LEP (Yes)			−0.017
			(0.361)
Grade (1)			2.395***
			(0.335)
Grade (2)			4.216***
			(0.341)
IEP (Yes)			−1.120**
			(0.427)
Race (American Indian or Alaska Native)			1.229*
			(0.604)
Race (Asian)			2.376***
			(0.608)
Race (Black or African American)			0.453
			(0.630)
Race (Hispanic)			−0.308
			(0.377)
Race (Mixed or Other)			1.201+
			(0.724)
SD (Intercept Child ID)	2.633	2.647	1.990
SD (Observations)	5.222	5.201	5.187
R^2 Marginal	0.000	0.005	0.095
R^2 Conditional	0.203	0.210	0.211
AIC	13239.0	13233.0	13078.1
BIC	13255.9	13255.6	13168.3
ICC	0.20	0.21	0.13
RMSE	4.76	4.73	4.88

Note. Likelihood Ratio Test between Time as a Random Slope and Intercept Only Models = $\chi^2(13) = 186.87$, $p < .001$; LEP: limited English proficiency; IEP: Individualized Education Program; Race reference group = White; $^+ p < 0.1$, $^* p < 0.05$, $^{**} p < 0.01$, $^{***} p < 0.001$.

explanatory power (Marginal $R^2 = 0.095$, Conditional $R^2 = 0.211$). The Intraclass Correlation Coefficient (ICC) was 0.13, suggesting that 13% of the variance in the outcome variable is attributable to differences between time points. These findings suggest that the rubric demonstrates reliability across time points, with demographic variables contributing to the understanding of score variations.

Discussion

This study aimed to develop and validate a creative coding rubric to assess early elementary children's CS skills and creative expression through an artifact-based assessment. The research questions guided our exploration of the rubric's validity, reliability, and its application over time. Our aim was to contribute to the small, but growing field, of early childhood coding assessments, specifically through emphasizing the expressive and creative nature of CS (Girshin et al., 2023; Kafai & Burke, 2014; Macrides et al., 2022). We did so by drawing upon previous work (Unahalekhaka & Bers, 2022), to create a rubric which can evaluate children's creative coding skills across different programming environments.

Validity and reliability of the rubric (RQ1)

Our investigation into the rubric's validity involved rigorous analysis, including inter-rater reliability, Pearson correlation, and factor analysis. Our analysis revealed substantial agreement among raters. In particular, we observed strong agreement for higher proficiency categories, affirming the rubric's ability to differentiate between varying levels of coding skills and creative expression. Moreover, we found the rubric to be robust and reliable, exemplifying the rubric's utility in future quantitative studies of young children's coding abilities, an area of research which is in need of attention (Pila et al., 2019).

Further, we sought to create a rubric which reflected the established relationship between coding and creativity (Sun et al., 2023; Wang et al., 2023). The convergence of findings across different analytical approaches provides robust evidence that the creative coding rubric captures a unified underlying construct of coding proficiency and creative expression.

In this paper, we argue that coding proficiency and creative thinking cannot happen independently of one another. Coding is a form of creation, thus the two are interconnected processes which both strengthen the other, aligning with previous literature discussing the overlap of the two processes (Kafai & Burke, 2014). A primary example of this would be the broadcasting function in ScratchJr—the act of triggering one program through another. Broadcasting, without an application, is often a difficult concept for children to grasp. However, with expressive applications such as coding a conversation between characters in service of telling a story, children naturally explore the advanced broadcasting feature in the interest of their creative expression. This demonstrates how the skills of coding advancement and creative expression can only help one another, and thus be treated as an interrelated set of skills.

This approach is consistent with the acknowledgment of coding's expressive potential in fostering creative thinking and problem solving abilities (Kafai & Burke, 2014). By integrating open-ended exploration and project-based learning into the assessment framework, our rubric provides a solution to a longstanding challenge faced by early childhood educators: seamlessly integrating assessment into play-based pedagogies (DeLuca et al., 2020a; Gullo & Hughes, 2011; Jeynes, 2006; Lynch, 2015; Pyle et al., 2022).

Reliability of the rubric over time (RQ2)

To investigate the rubric's reliability over time, we employed HLM to examine consistency in scores across multiple time points. Our analysis revealed that including time as a predictor significantly enhanced model fit, indicating that changes in elementary children's coding proficiency over time were better captured by the expanded model incorporating demographic factors. This demonstrates the rubric's utility not only as a creative coding measure, but as a creative coding measure over time. Given the push for accountability in early childhood education as well as the lack of quantitative research around early childhood coding, this rubric addresses a critical need among educators and researchers alike (DeLuca et al., 2020a; Pila et al., 2019).

By proposing an artifact-based assessment for open-source creative coding platforms, we aim to provide educators, researchers, and policymakers with a way to evaluate children's learning beyond the confines of traditional standardized testing. This approach aligns with the effectiveness of project-based learning in assessing coding skills and promotes a shift toward recognizing the expressive potential of coding in fostering creative thinking (Basu, 2019; Grover et al., 2018; Unahalekhaka & Bers, 2022). Through introducing this type of measurement, we hope that the focus of school achievement and curricular standards can be expanded to include a stronger emphasis on the creative process of learning.

Implications for education and future research

The validation and reliability of the Creative Coding Rubric presented in this study have implications for both education and future research. In educational settings, our rubric provides a valuable tool for educators to assess and nurture the creative coding skills of elementary children. The emphasis on creative expression aligns with contemporary educational goals aimed at cultivating well-rounded individuals capable of innovative thinking and problem solving (DeLuca et al., 2020a). By evaluating coding projects, educators can gain valuable insights into children's learning progression and identify areas for targeted instruction and support.

Moreover, although the rubric was developed and validated for ScratchJr, the rubric's adaptability to various programming environments makes it versatile for use in different educational contexts. Educators can leverage it to tailor coding assessments to the specific programming tools and languages used in their classrooms. We offer suggestions for ways the rubric can be adapted in the Appendix in the Foos Adaptation section. However, since ScratchJr currently is used by over 50 million users worldwide, this rubric will provide immediately applicable support to educators and researchers due to the reach and impact that ScratchJr currently has in classrooms and homes worldwide.

While our study sets the stage for more in-depth exploration of the relationship between creative coding skills and broader outcomes in children, we acknowledge a series of limitations with our work that we hope will be addressed in future research. The present research primarily worked with participants from two states in the northeast region of the United States, indicating a need for greater global representation in future work. Furthermore, the sample was not equally distributed across race, SES, IEP, or LEP status, and we believe our findings can be strengthened through working with a more diverse sample. The rubric's reliability over time suggests its potential usability in longitudinal studies tracking elementary children's coding development. Future research could explore the applicability of the rubric across diverse populations and programming environments, including formal educational settings and informal learning contexts, over time. Further validation studies could also investigate the generalizability of the rubric across different age groups and programming languages, enhancing its versatility and utility in assessing coding proficiency and creativity.

In summary, the Creative Coding Rubric's adaptability makes it a valuable tool for researchers and educators seeking to understand children's natural learning progression in creative coding and positions it as a versatile assessment tool across various contexts. Encouraging further exploration of the rubric's validity in new settings, we envision its role in advancing our understanding of children's creative coding development on a global scale. Aligning with broader efforts to develop robust tools for evaluating elementary children's CS skills and creative expression, our aim is to propel the field's comprehension of how creative coding education fosters both creativity and computational thinking in children.

Notes

1. Creative Computing Curriculum (v1. 2014, v2. 2019) was developed by the Creative Computing Lab at Harvard University and was translated into 12 languages. To view the curriculum, visit https://creativecomputing.gse.harvard.edu/guide. Adventures in Alice Programming was developed at Duke University to support all ranges of Alice. For more information visit http://www.alice.org/resources/exercise-and-project/adventures-in-alice-programming/.
2. To learn more, visit https://www.scratchjr.org/about/info
3. Cohen's standard indicates that coefficients between .10 and .29 represent a small effect size, coefficients between .30 and .49 represent a moderate effect size, and coefficients above .50 indicate a large effect size
4. Where > .9 excellent, > .8 good, > .7 acceptable, > .6 questionable, > .5 poor, and ≤ .5 unacceptable.
5. Any Mahalanobis distance exceeding 22.46, corresponding to the .999 quantile of a χ^2 distribution with 6 degrees of freedom, was considered influential (Kline, 2015).
6. In a sensitivity analysis, *State* was included in the model. However, its addition resulted in a worse model fit and it was not found significant.
7. Generally, Fleiss' Kappa coefficients can be interpreted as follows: < 0: Poor agreement; 0.01 - 0.20: Slight agreement; 0.21 - 0.40: Fair agreement; 0.41 - 0.60: Moderate agreement; 0.61 - 0.80: Substantial agreement; 0.81 – 1.000: Almost perfect agreement (Landis & Koch, 1977).

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This paper is based on work supported by the Department of Education under Grant Number U411C220202. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Department of Education.

Notes on contributors

Jessica Blake-West is a second year doctoral student as well as the ScratchJr Learning Experience Designer for the Developmental Technologies (DevTech) research group at Boston College. She completed her Bachelor's of Science in Cognitive Neuroscience at Brown University and her Master's of Science in Human Factors Engineering at Tufts University. Her interests include researching and designing technologies which foster creativity and collaboration for young children.

Ghaida Alrawashdeh is an assistant professor of digital learning and literacies at Oakland University. She uses educational technology to support learning and teaching in resource-challenged contexts.

Marina Bers is the Augustus Long Professor of Education at the Lynch School of Education and Human Development at Boston College. Prof. Bers heads the interdisciplinary Developmental Technologies (DevTech) research group at Boston College. Her research involves the design and study of innovative learning technologies to promote children's positive development.

ORCID

Jessica Blake-West http://orcid.org/0000-0003-2109-0503
Ghaida Alrawashdeh http://orcid.org/0000-0003-4017-1085
Marina Bers http://orcid.org/0000-0003-0206-1846

References

Al Kandari, A., & Al Qattan, M. (2020). E-task-based learning approach to enhancing 21st-century learning outcomes. *International Journal of Instruction*, *13*(1), 551–566. https://doi.org/10.29333/iji.2020.13136a

Angeli, C., & Valanides, N. (2020). Developing young children's computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy. *Computers in Human Behavior*, *105*, 105954. https://doi.org/10.1016/j.chb.2019.03.018

Barron, B., & Darling-Hammond, L. (2008). Teaching for meaningful learning: A review of research on inquiry-based and cooperative learning. In *George Lucas educational foundation*. George Lucas Educational Foundation. https://eric.ed.gov/?id=ED539399

Barron, B., Bofferding, L., Cayton-Hodges, G., Copple, C., Darling-Hammond, L., & Levine, M. H. (2011). *Take a giant step: A blueprint for teaching young children in a digital age*. The Joan Ganz Cooney Center at Sesame Workshop. https://joanganzcooneycenter.org/wp-content/uploads/2012/01/jgcc_takeagiantstep1.pdf

Basu, S. (2019). *Using rubrics integrating design and coding to assess middle school students' open-ended block-based programming projects* [Paper presentation]. Proceedings of the 50th ACM Technical Symposium on Computer Science Education, 1211–1217. https://doi.org/10.1145/3287324.3287412

Basu, S., McElhaney, K. W., Grover, S., Harris, C. J., Biswas, G. (2018). *A principled approach to designing assessments that integrate science and computational thinking*. 13th International Conference of the Learning Sciences (ICLS), 1.

Bates, D., Maechler, M., Bolker, B., Walker, S., Christensen, R. H. B., Singmann, H., Dai, B., Scheipl, F., Grothendieck, G., Green, P., Fox, J., Bauer, A., Krivitsky, P. N., Tanaka, E. (2023). *lme4: Linear Mixed-Effects Models using "Eigen" and S4* (1.1–35.1) [Computer software]. https://cran.r-project.org/web/packages/lme4/index.html

Bati, K. (2022). A systematic literature review regarding computational thinking and programming in early childhood education. *Education and Information Technologies*, *27*(2), 2059–2082. https://doi.org/10.1007/s10639-021-10700-2

Ben Ari, A., Levinson, T., Bers, M. U., & Rosenberg-Kima, R. (2023). Nurturing computational thinking in an Israeli kindergarten with the CAL-KIBO robotics curriculum [Paper presentation]. Aera 2023. AERA 2023. https://doi.org/10.3102/IP.23.2012601

Bers, M. (2021). *Teaching computational thinking and coding to young children*. IGI Global.

Bers, M. U., Blake-West, J., Kapoor, M. G., Levinson, T., Relkin, E., Unahalekhaka, A., & Yang, Z. (2023). Coding as another language: Research-based curriculum for early childhood computer science. *Early Childhood Research Quarterly*, *64*, 394–404. https://doi.org/10.1016/j.ecresq.2023.05.002

Booton, S. A., Kolancali, P., & Murphy, V. A. (2023). Touchscreen apps for child creativity: An evaluation of creativity apps designed for young children. *Computers & Education*, *201*, 104811. https://doi.org/10.1016/j.compedu.2023.104811

Bortz, W. W., Gautam, A., Tatar, D., & Lipscomb, K. (2020). Missing in measurement: Why identifying learning in integrated domains is so hard. *Journal of Science Education and Technology*, *29*(1), 121–136. https://doi.org/10.1007/s10956-019-09805-8

Broadhead, P. (2006). Developing an understanding of young children's learning through play: The place of observation, interaction and reflection. *British Educational Research Journal*, *32*(2), 191–207. https://doi.org/10.1080/01411920600568976

Brown, N. C. C., Sentance, S., Crick, T., & Humphreys, S. (2014). Restart: The resurgence of computer science in UK schools. *ACM Transactions on Computing Education*, *14*(2), 1–22. https://doi.org/10.1145/2602484

Canbeldek, M., & Isikoglu, N. (2023). Exploring the effects of "productive children: Coding and robotics education program" in early childhood education. *Education and Information Technologies*, *28*(3), 3359–3379. https://doi.org/10.1007/s10639-022-11315-x

Catete, V., Snider, E., & Barnes, T. (2016). *Developing a rubric for a creative cs principles lab* [Paper presentation]. (p. 295). https://doi.org/10.1145/2899415.2899449

Chou, P.-N. (2020). Using Scratchjr to foster young children's computational thinking competence: A case study in a third-grade computer class. *Journal of Educational Computing Research*, *58*(3), 570–595. https://doi.org/10.1177/0735633119872908

Chun, H. Y., & Park, S. (2020). The effect of steam-based unplugged play activities using robots on the improvement of children's creative and social personalities. *Korean Journal of Childcare and Education*, *16*(5), 1–26. https://doi.org/10.14698/JKCCE.2020.16.05.001

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Routledge. https://doi.org/10.4324/9780203771587

Costello, A. B., & Osborne, J. (2005). *Best practices in exploratory factor analysis: Four recommendations for getting the most from your analysis*. https://doi.org/10.7275/JYJ1-4868

Cutumisu, M., Adams, C., & Lu, C. (2019). A scoping review of empirical research on recent computational thinking assessments. *Journal of Science Education and Technology*, *28*(6), 651–676. https://doi.org/10.1007/s10956-019-09799-3

de Ruiter, L. E., & Bers, M. U. (2022). The coding stages assessment: Development and validation of an instrument for assessing young children's proficiency in the ScratchJr programming language. *Computer Science Education*, *32*(4), 388–417. https://doi.org/10.1080/08993408.2021.1956216

DeLuca, C., Pyle, A., Braund, H., & Faith, L. (2020a). Leveraging assessment to promote kindergarten learners' independence and self-regulation within play-based classrooms. *Assessment in Education-Principles Policy & Practice*, *27*(4), 394–415. https://doi.org/10.1080/0969594X.2020.1719033

DeLuca, C., Pyle, A., Valiquette, A., & LaPointe-McEwan, D. (2020b). New directions for kindergarten education: Embedding assessment in play-based learning. *The Elementary School Journal*, *120*(3), 455–479. https://doi.org/10.1086/707008

Denner, J., Werner, L., & Ortiz, E. (2012). Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts? *Computers & Education*, *58*(1), 240–249. https://doi.org/10.1016/j.compedu.2011.08.006

Di Lieto, M. C., Inguaggiato, E., Castro, E., Cecchi, F., Cioni, G., Dell'Omo, M., Laschi, C., Pecini, C., Santerini, G., Sgandurra, G., & Dario, P. (2017). Educational Robotics intervention on Executive Functions in preschool children: A pilot study. *Computers in Human Behavior*, *71*, 16–23. https://doi.org/10.1016/j.chb.2017.01.018

Falkner, K., Vivian, R., & Falkner, N. (2014). *The Australian digital technologies curriculum: Challenge and opportunity* (vol. *148*, pp. 3–12).

Field, A. (2017). *Discovering statistics using IBM SPSS statistics: North American edition* (5th ed.). SAGE Publications Ltd.

Flannery, L. P., Silverman, B., Kazakoff, E. R., Bers, M. U., Bontá, P., & Resnick, M. (2013). *Designing ScratchJr: Support for early childhood learning through computer programming* [Paper presentation]. Proceedings of the 12th International Conference on Interaction Design and Children, 1–10. https://doi.org/10.1145/2485760.2485785

Fragapane, V., & Standl, B. (2021). *Work in progress: Creative coding and computer science education – from approach to concept* [Paper presentation].2021 IEEE Global Engineering Education Conference (EDUCON), 1233–1236. https://doi.org/10.1109/EDUCON46332.2021.9453951

Gamer, M., Lemon, J., Singh, I. F. P. (2019). *Irr: Various coefficients of interrater reliability and agreement* (0.84.1) [Computer software]. https://cran.r-project.org/web/packages/irr/index.html

George, D., & Mallery, P. (2019). Reliability analysis. In *IBM SPSS Statistics 26 Step by Step* (16th ed.). Routledge.

Girshin, R., Rosenberg, N., & Kukliansky, I. (2023). Early childhood robotics: Children's beliefs and objective capabilities to read and write programs. *Journal of Research in Childhood Education*, *38*(2), 317–335. https://doi.org/10.1080/02568543.2023.2259946

Goldstein, J., McCoach, D. B., & Yu, H. (2017). The predictive validity of kindergarten readiness judgments: Lessons from one state. *The Journal of Educational Research*, *110*(1), 50–60. https://doi.org/10.1080/00220671.2015.1039111

Grover, S. (2020). Designing an assessment for introductory programming concepts in middle school computer science [Paper presentation]. Proceedings of the 51st ACM Technical Symposium on Computer Science Education, 678–684. https://doi.org/10.1145/3328778.3366896

Grover, S., & Pea, R. (2013). Computational thinking in k–12: A review of the state of the field. *Educational Researcher*, *42*(1), 38–43. https://doi.org/10.3102/0013189X12463051

Grover, S., Basu, S., & Schank, P. (2018). *What we can learn about student learning from open-ended programming projects in middle school computer science* [Paper presentation]. Proceedings of the 49th ACM Technical Symposium on Computer Science Education, 999–1004. https://doi.org/10.1145/3159450.3159522

Gullo, D. F., & Hughes, K. (2011). Reclaiming kindergarten: Part I. questions about theory and practice. *Early Childhood Education Journal*, *38*(5), 323–328. https://doi.org/10.1007/s10643-010-0429-6

Hendrickson, K., Gauthier, L., Glennon, M. O., Harrigan, A. M., Weissman, H., Fletcher, C., Dunton, S., Baskin, J., Mak, J. (2021). *2021 computer science education policy*. Code.org, CSTA, & ECEP Alliance. https://advocacy.code.org/stateofcs

Hirsh-Pasek, K., Hadani, H. S., Blinkoff, E., Golinkoff, R. (2020). *A new path to education reform: Playful learning promotes 21st-century skills in schools and beyond* (Policy 2020). The Brookings Institution. https://www.brookings.edu/wp-content/uploads/2020/10/Big-Ideas_Hirsh-Pasek_PlayfulLearning.pdf

Hooper, D., Coughlan, J., & Mullen, M. (2008). Structural equation modeling: Guidelines for determining model fit. *Articles*. https://doi.org/10.21427/D7CF7R

Hu, L. (2023). Programming and 21st century skill development in K-12 schools: A multidimensional meta-analysis. *Journal of Computer Assisted Learning*, *40*(2), 610–636. https://doi.org/10.1111/jcal.12904

Ioannou, A., & Makridou, E. (2018). Exploring the potentials of educational robotics in the development of computational thinking: A summary of current research and practical proposal for future work. *Education and Information Technologies*, *23*(6), 2531–2544. https://doi.org/10.1007/s10639-018-9729-z

Jeynes, W. H. (2006). Standardized tests and froebel's original kindergarten model. *Teachers College Record: The Voice of Scholarship in Education*, *108*(10), 1937–1959. https://doi.org/10.1111/j.1467-9620.2006.00769.x

Kafai, Y. B., & Burke, Q. (2014). *Connected code: Why children need to learn programming*. The MIT Press.

Kim, Y., & Jeong, H. (2023). Adopting coding in early childhood education: Teachers' perspectives. *IEEE Transactions on Learning Technologies*, *16*(4), 473–483. https://doi.org/10.1109/TLT.2023.3264681

Kline, R. B. (2015). *Principles and practice of structural equation modeling* (4th ed.). The Guilford Press.

Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, *33*(1), 159–174. https://doi.org/10.2307/2529310

Lee, A. L., & Comrey, H. B. (2013). *A first course in factor analysis* (2nd ed.). Psychology Press. https://doi.org/10.4324/9781315827506

Lynch, M. (2015). Ontario kindergarten teachers' social media discussions about full day kindergarten. *McGill Journal of Education*, *49*(2), 329–347. https://doi.org/10.7202/1029423ar

Macrides, E., Miliou, O., & Angeli, C. (2022). Programming in early childhood education: A systematic review. *International Journal of Child-Computer Interaction*, *32*, 100396. https://doi.org/10.1016/j.ijcci.2021.100396

Monteiro, A. F., Miranda-Pinto, M., & Osório, A. J. (2021). Coding as literacy in preschool: A case study. *Education Sciences*, *11*(5), 198. https://doi.org/10.3390/educsci11050198

Montuori, C., Gambarota, F., Altoé, G., & Arfé, B. (2024). The cognitive effects of computational thinking: A systematic review and meta-analytic study. *Computers & Education*, *210*, 104961. https://doi.org/10.1016/j.compedu.2023.104961

Moreno León, J., Robles, G., & Román-González, M. (2016). Code to learn: Where does it belong in the K-12 curriculum? *Journal of Information Technology Education: Research*, *15*, 283–303. https://doi.org/10.28945/3521

Moreno-León, J., & Robles, G. (2015). *Dr. Scratch: A web tool to automatically evaluate Scratch projects* [Paper presentation]. Proceedings of the Workshop in Primary and Secondary Computing Education, 132–133. https://doi.org/10.1145/2818314.2818338

Mouza, C., Codding, D., & Pollock, L. (2022). Investigating the impact of research-based professional development on teacher learning and classroom practice: Findings from computer science education. *Computers & Education*, *186*, 104530. https://doi.org/10.1016/j.compedu.2022.104530

Nouri, J., Zhang, L., Mannila, L., & Norén, E. (2020). Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. *Education Inquiry*, *11*(1), 1–17. https://doi.org/10.1080/20004508.2019.1627844

Papadakis, S., Kalogiannakis, M., & Zaranis, N. (2016). Developing fundamental programming concepts and computational thinking with ScratchJr in preschool education: A case study. *International Journal of Mobile Learning and Organisation*, *10*(3), 187. https://doi.org/10.1504/IJMLO.2016.077867

Papavlasopoulou, S., Sharma, K., & Giannakos, M. N. (2020). Coding activities for children: Coupling eye-tracking with qualitative data to investigate gender differences. *Computers in Human Behavior*, *105*, 105939. https://doi.org/10.1016/j.chb.2019.03.003

Pila, S., Aladé, F., Sheehan, K. J., Lauricella, A. R., & Wartella, E. A. (2019). Learning to code via tablet applications: An evaluation of Daisy the Dinosaur and Kodable as learning tools for young children. *Computers & Education*, *128*, 52–62. https://doi.org/10.1016/j.compedu.2018.09.006

Popat, S., & Starkey, L. (2019). Learning to code or coding to learn? A systematic review. *Computers & Education*, *128*, 365–376. https://doi.org/10.1016/j.compedu.2018.10.005

Pyle, A., DeLuca, C., Danniels, E., & Wickstrom, H. (2020). A model for assessment in play-based kindergarten education. *American Educational Research Journal*, *57*(6), 2251–2292. https://doi.org/10.3102/0002831220908800

Pyle, A., DeLuca, C., Wickstrom, H., & Danniels, E. (2022). Connecting kindergarten teachers? Play-based learning profiles and their classroom assessment practices. *Teaching and Teacher Education*, *119*, 103855. https://doi.org/10.1016/j.tate.2022.103855

Relkin, E., de Ruiter, L., & Bers, M. U. (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. (2017). *Lifelong kindergarten: Cultivating creativity through projects, passion, peers, and play*. MIT press. https://books.google.com/books?hl=en&lr=&id=nGYyDwAAQBAJ&oi=fnd&pg=PR5&dq=info:0WM2rkjRzzMJ:scholar.google.com&ots=lD4ygv2QvA&sig=Qkspqc7zgWciY58W_s-7OQUyxN0

Sharmin, S. (2021). Creativity in cs1: A literature review. *ACM Transactions on Computing Education*, *22*(2), 1–26. https://doi.org/10.1145/3459995

Sherman, M., & Martin, F. (2015). The assessment of mobile computational thinking. *Journal of Computing Sciences in Colleges*, *30*, 53–59.

Su, J., Yang, W., & Zhong, Y. (2023). Influences of gender and socioeconomic status on children's use of robotics in early childhood education: A systematic review. *Early Education and Development*, *34*(4), 910–926. https://doi.org/10.1080/10409289.2022.2078617

Sun, L., & Zhou, D. (2023). Effective instruction conditions for educational robotics to develop programming ability of K-12 students: A meta-analysis. *Journal of Computer Assisted Learning*, *39*(2), 380–398. https://doi.org/10.1111/jcal.12750

Sun, L., Guo, Z., & Zhou, D. (2023). Measuring development of young students' coding ability through a graphical teaching intervention: Further explanation of the effect of coding experience and coding interest. *Interactive Learning Environments*, 1–24. https://doi.org/10.1080/10494820.2023.2190356

Sung, W., Ahn, J., & Black, J. B. (2017). Introducing computational thinking to young learners: Practicing computational perspectives through embodiment in mathematics education. *Technology, Knowledge and Learning*, *22*(3), 443–463. https://doi.org/10.1007/s10758-017-9328-x

Tabachnick, B., Fidell, L. (2019). *Using multivariate statistics* (7th ed.). Pearson. https://www.amazon.com/Using-Multivariate-Statistics-Barbara-Tabachnick/dp/0134790545

Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education*, *148*, 103798. https://doi.org/10.1016/j.compedu.2019.103798

The White House. (n.d.). *The Biden-Harris administration immediate priorities*. The White House. https://www.whitehouse.gov/priorities/

Tselegkaridis, S., & Sapounidis, T. (2022). Exploring the features of educational robotics and stem research in primary education: A systematic literature review. *Education Sciences*, *12*(5), 305. https://doi.org/10.3390/educsci12050305

Unahalekhaka, A., & Bers, M. U. (2022). Evaluating young children's creative coding: Rubric development and testing for ScratchJr projects. *Education and Information Technologies*, *27*(5), 6577–6597. https://doi.org/10.1007/s10639-021-10873-w

Wakil, K., Khdir, S., Sabir, L., & Nawzad, L. (2019). Student ability for learning computer programming languages in primary schools. *International e-Journal of Educational Studies*, *3*(6), 109–115. https://doi.org/10.31458/iejes.591938

Wang, L., Geng, F., Hao, X., Shi, D., Wang, T., & Li, Y. (2023). Measuring coding ability in young children: Relations to computational thinking, creative thinking, and working memory. *Current Psychology*, *42*(10), 8039–8050. https://doi.org/10.1007/s12144-021-02085-9

Weintrop, D., Beheshti, E., Horn, M. S., Orton, K., Trouille, L., Jona, K., & Wilensky, U. (2014). Interactive assessment tools for computational thinking in high school STEM classrooms. In D. Reidsma, I. Choi, & R. Bargar (Eds.), *Intelligent technologies for interactive entertainment* (vol. *136*, pp. 22–25). Springer International Publishing. https://doi.org/10.1007/978-3-319-08189-2_3

Weissman, H., Glennon, M. O., Twarek, B., Dunton, S., & Childs, J. (2023). *2023 state of computer science education*. Code.org, CSTA, ECEP Alliance. https://advocacy.code.org/2023_state_of_cs.pdf

Wilson, A., Hainey, T., & Connolly, T. M. (2013). Using Scratch with primary school children: An evaluation of games constructed to gauge understanding of programming concepts. *International Journal of Game-Based Learning*, *3*(1), 93–109. https://doi.org/10.4018/ijgbl.2013010107

Yang, D., Yang, Z., & Bers, M. U. (2023). The efficacy of a computer science curriculum for early childhood: Evidence from a randomized controlled trial in K-2 classrooms. *Computer Science Education*, 1–21. https://doi.org/10.1080/08993408.2023.2279198

Yang, Z., & Bers, M. U. (2023). Examining gender difference in the use of ScratchJr in a programming curriculum for first graders. *Computer Science Education*, 1–22. https://doi.org/10.1080/08993408.2023.2224135