



Test for assessing coding skills in early childhood

Damla Kalyenci¹ · Şermin Metin² · Mehmet Başaran³

Received: 27 July 2021 / Accepted: 3 November 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

This research aims to develop a valid and reliable test to determine the coding skill levels of 5–7 years old children in early childhood. The study sample consists of children aged 5–7 who attend primary and pre-school education institutions affiliated to the Ministry of National Education in Ağrı and Gaziantep city center in the 2020–2021 academic year. Data were obtained from 308 children, 101 of whom were five years old, 100 were six years old, and 107 were seven. As a data collection tool in research, the “Personal Information Form” containing personal information about children and their parents and the “Early Childhood Coding Skills Assessment Test” developed by the researcher to evaluate the coding skill levels of 5–7-year-old children were used. In the validity analysis to determine the test’s validity and reliability, content-structure validity, criterion-based validity analysis, similar scale compatibility validity, tetrachoric factor analysis, and item difficulty analysis; In the reliability analysis, KR-20 reliability analysis was used. As a result of the findings obtained from the research, the “Early Childhood Coding Skills Assessment Test” is a valid and reliable measurement tool that can be used to determine the skill levels of 5–7-year-old children unplugged coding and robotic coding.

Keywords Early childhood · Assessment · Unplugged coding · Robotic coding

✉ Mehmet Başaran
mehmetbasaran@outlook.com

Damla Kalyenci
zdamlakalyenci@hotmail.com

Şermin Metin
sermin.metin@hku.edu.tr

¹ Expert, Ministry of National Education, Ağrı, Turkey

² Faculty of Education, Department of Preschool Teacher Education, Hasan Kalyoncu University, Gaziantep, Turkey

³ Faculty of Education, Department of Educational Sciences, Gaziantep University, Gaziantep, Turkey

1 Introduction

Children live in automated systems based on coding in their daily lives and grow up saturated with smart devices. Due to rapid changes in technology, children are increasingly exposed to these systems, and this exposure naturally increases their interest in how objects function or move automatically. The ubiquity of technology positions it as an important cultural tool for children to understand the world. Therefore, the need for a workforce that understands technology in the future has made coding and computational thinking a training priority of national education programs (Lee & Junoh, 2019; Johnston et al., 2018; Campbell & Walsh, 2017; Manches & Plowman, 2017; Rogoff et al., 1995).

Children learn human languages for speaking and writing, natural languages covering all subjects related to experimental sciences (physics, chemistry, biology, etc.), and human languages including social sciences and humanities, as well as digital languages, one of the necessary competencies for children to be successful in the digital world. is required (Briggs, 2013; García Peñalvo et al., 2016). Within the framework of twenty-first-century skills, to successfully and actively participate in a digital society, all children must acquire basic digital literacy skills, including information literacy, media literacy, and information, communication, and technology (ICT) literacy (P21, 2017).

Children need to meet with essential digital literacy and coding, which is seen as the digital language early. Because early childhood years are considered as a crucial critical time in the development of coding and programming skills as well as in all skills (Flannery & Bers, 2013; Campbell & Walsh, 2017; Ananiadou & Claro, 2009). In recent years, efforts have been directed towards teaching children basic skills called specific computer science concepts, programming skills, and computational thinking skills (Relkin et al., 2020). Coding, a new literacy, has become a fundamental tool for reading, interpreting, and communicating with others in a digital society, providing children with the opportunity to connect with technology. Beyond algorithmic thinking, coding offers a symbolic language in which children will have reading and writing skills (Bers, 2018b, a; McLennan, 2017).

2 Coding / robotic coding

Considering the definitions related to coding, which is defined as a new literacy of the twenty-first century, coding; is defined as the process of writing the correct syntax regularly and sequentially and developing applications by using command sets in order to solve problems, provide human-computer interaction and perform a specific task by the computer (Vorderman, 2019; Bers et al., 2019; McLennan, 2018; Kalelioğlu et al., 2016; Demirer & Sak, 2016; Fesakis & Serafeim, 2009; Wing, 2006). It is a set of instructions used to define each step to perform a specific task or solve a specific problem and includes an algorithm design

(Lee & Björklund Larsen, 2019); thinking style in coding; Numerical thinking is seen as a process of solving problems using algorithms and developing a logical approach, analyzing and organizing data, dividing problems into small and manageable parts, converting them into specific algorithms and converting them into programming languages (Bers et al., 2019; Sullivan et al., 2017; Gibson, 2012; Futschek & Moschitz, 2011; Arabacıoğlu et al., 2007; Futschek, 2006).

There is no difference between these two concepts, which are considered coding or programming, and both concepts mean giving instructions to a machine or robot. Coding can also be seen as the result of algorithms, a list of step-by-step procedures to complete a particular task (Campbell & Walsh, 2017; Lee & Junoh, 2019). While coding is a relatively new term in early childhood education, children experience and use coding in their daily lives and routines through wireless applications, such as learning to tie shoelaces by following a series of steps (Lee & Junoh, 2019; Resnick & Siegel, 2015; Sullivan & Bers, 2016; Ananiadou & Claro, 2009). However, Resnick and Siegel (2015) state that introductory coding activities do not allow children to express themselves creatively or develop a long-term interaction with coding. Similarly, Bers et al. (2019) emphasizes that crossword-type approaches to coding skills will cause children to miss the opportunity to explore the richness of a programming language and coding as a symbol system that includes grammar and syntax.

Coding, which is accepted as a new language, should start with a simple order like acquiring the language (Bers, 2019). In early childhood, basic skills in coding include knowing code concepts and code cards (Futschek, 2006; Futschek & Moschitz, 2010; Gibson, 2012; Mittermeir, 2013; Sullivan & Bers, 2016). The coding process includes the skills of direction signs, sequencing, debugging, creating functions, loops, and algorithmic thinking (Lee & Junoh, 2019; Zamin et al., 2018; Mittermeir, 2013; Gibson, 2012; Futschek & Moschitz, 2010; Futschek, 2006; Welch et al., 2019). Bers et al. (2014) state that the design process of the coding curriculum in early childhood should consist of solid ideas from computer science and engineering, which include debugging, robotic motion and sensing, using programming instructions, sequential and specific instructions flow control.

Coding skills and standards for these skills are set forth at the K–12 level by the Computer Science Teachers Association / CSTA (, 2003, 2011, 2019) and the International Society for Technology in Education / ISTE (2019). The coding skills revealed by CSTA, ISTE, and many researchers are listed in Table 1.

The Computer Science Teachers (CSTA) Standards are developed by the Computer Science Teachers Association (CSTA), a nonprofit organization dedicated to empowering, involving, and advocating computer science teachers in the preschool-to-college education (K-12) education process worldwide. It is written and maintained by teacher members of the Computer Science Teachers Association/CSTA. These Standards were first created and published as standards for Computer Science (CS) educators by the International Society for Technology in Education (ISTE) in 2003, last updated in 2011, and ISTE for publication in 2020. It was rewritten in 2019 in partnership with. The CSTA K-12 Computer Science Standards define core learning objectives designed to form the basis for a complete computer science curriculum and its implementation at the K-12 level. However, the ISTE Standards are a

Table 1 Coding process

1A-AP-09: While writing the program, children can use the up / down directions as yes / no statements, use arrows to represent the direction while writing algorithms	Signs (Start, end, down, up, right, left, loop)	The child is read the story about the question. The child is asked to code for the monkey to go to the banana.	
1A-AP-11: These are the instruction steps given while performing the algorithm steps.	Sorting	The child is read the story about the question. The child is asked to code the robot following the order in the story.	
1A-AP-10: Detection and correction of false statements and transactions in the algorithm	Debugging	The child is read the story about the question. The child is asked to code the robot so that the monkey can reach the result without getting stuck in obstacles.	
1A-AP-14: A structure that allows a sequence of code to be repeated multiple times.	Loops	The child is read the story about the question. The child is asked to code the robot following the specified loop rule for the monkey to take the bananas.	
1A-AP-11: Parsing is the act of breaking down tasks into more straightforward tasks	Modularity	The child is read the story about the question. The child is asked to code the tasks necessary for the robot monkey to reach the result.	
1A-AP-08: The algorithm combines smaller tasks into more complex ones.	Algorithm	The child is read the story about the question. The child is asked to code the robot according to the loop rule that will lead the monkey to the banana	
1A-AP-15: It is to create a plan of what a program will do.	Program development	The child is given the characters of the story related to the question. The child is asked to create a story using the pictures given below and to code the robot according to the story.	

*Descriptions and pictures here are from CodingTest B form

framework for innovation in education, and these standards help educators and education leaders around the world prepare students to thrive in work and life. Today's children are expected to be ready to thrive in an ever-evolving technological environment. Standards for ISTE students are designed to strengthen the student's voice and ensure that learning is a student-centered process (ISTE, 2016).

In the development process of technology, it is necessary to develop solutions for how the education they receive can be permanent and effective while introducing coding, which will affect the thinking skills of future engineers and software developers or in all areas of life. (Ananiadou & Claro, 2009; Ackermann, 2001). Coding,

which is ubiquitous in children's lives, needs to expose them in a developmentally appropriate manner. Developmental approaches that allow children to manipulate codes and do not require a computer are gaining importance. Using the computer and unplugged and activity-based applications and robotic tools and equipment while gaining coding skills is one of these solutions (Metin, 2020; Sullivan & Bers, 2016; Bers et al., 2014; Sullivan et al., 2013; Wang et al., 2011; Futschek & Moschitz, 2010). While programming can be seen as an abstract activity in itself, physical objects such as robots make the activity more concrete while also encouraging collaboration around the programming task at hand (Bers et al., 2019; Bers, 2018a, 2018b; Campbell & Walsh, 2017; Levy & Mioduser, 2008; Reese et al., 2019; Bell et al., 2012; Resnick & Siegel, 2015; Sullivan & Bers, 2016). There are two standard practices for improving coding skills: Computer does not require the use (as [code.org](#)) and is carried out using computer applications (Text, 2020; Bers et al., 2019; Bell et al., 2012). The concept of computer science unplugged, which embodies developmentally appropriate and abstract concepts for computer science and children's basic skills, has been accepted by many researchers and has been used in coding education (Metin, 2020; Bers, 2018a; Bell & Vahrenhold, 2018; Bell et al., 2012; Wang et al., 2011). In unplugged coding education, it seems more appropriate to teach coding through activities rather than computers, which are very complex tools for children to understand (Relkin et al., 2020; Metin, 2020; Bell & Vahrenhold, 2018; Bers et al., 2014; Gujberova & Kalas, 2013; Kazakoff et al., 2013; Bell et al., 2012; Futschek & Moschitz, 2010).

Children can learn best in meaningful ways and fun, based on practice and interaction (Fleer, 2013). Studies on this subject have revealed that coding education is more successful when it is carried out with concrete materials and experiences based on activities and active participation of children (Metin, 2020; Lee & Junoh, 2019; Wang et al., 2011; Campbell & Walsh, 2017; Levy & Mioduser, 2008; Bate-man et al., 2017); Bell et al., 2012; Resnick & Siegel, 2015; Sullivan & Bers, 2016). Piaget (1964) emphasizes that it is difficult for children in the pre-operational period to understand and construct abstract concepts without concrete experience. According to Piaget (1970) and Vygotsky (1980), learning in children is based on the active participation of the child and concrete experiences with adults and their peers. The first effort for coding training to be based on concrete experiences is Papert's (1980) structuralist approach based on Piaget's cognitive theory. Based on this idea, it is emphasized that the programming of many studies should depend on the physical environment and objects for concrete programming (Montemayor, 2003; Horn & Jacob, 2007; Scharf et al., 2008; McNerney, 2004).

Bers (2010, 2012) provided a framework to guide the implementations for coding with the Positive Technological Development (PTD) approach, which guided child-centered coding training. Today, the Positive Technological Development (PTD) framework, the basic coding and computational thinking skills approach, includes six C: communication, collaboration, community building, content creation, creativity, and behavior options (Bers, 2010, 2012; Horn et al., 2012). This approach (PTD) considers the learning environment and pedagogical practices, and cultural values and rituals that mediate teaching and learning (Bers, 2010; Rogoff et al., 2001). Game-based digital learning (PBDL) (Campbell and Walsh 2017; Futschek

& Moschitz, 2010), which forms the basis for coding training and requires the concretization of the digital world by playing developmentally, and coding in activity-based approaches that provide a practical experience for learners and facilitate learning (Metin, 2020; Sullivan & Bers, 2016; Kamenetz, 2016; Lijanporn & Khlaisang, 2015; Bers et al., 2014; Kazakoff et al., 2013; Lee et al., 2013; Wang et al., 2011; Stoeckelmayr et al., 2011; Futschek & Moschitz, 2010; Biazak et al., 2010; Scharf et al., 2008). Approaches to coding education in early childhood and compliance with the development emphasized by the studies and especially suggested by NAECY, CSTA (2020), and ISTE (2016) standards should also be considered. Cejka et al. (2006) emphasized that by combining coding and design with the help of computers, robotics, and Unplugged activities, students of all ages can provide a rich and meaningful learning experience.

3 Assessment of coding skills

As children's coding and robotic coding training become widespread, valid and reliable measurement tools are needed to evaluate these skills (Kalyenci, 2020). Many studies have been conducted on coding at the early childhood and primary education levels, and in these studies, the training has been evaluated through interviews, observations, and questionnaires. In studies conducted at the pre-school level, Bers et al. (2019) implemented robotic coding training with children aged 3–5 with different socioeconomic levels and 16 teachers. Data were analyzed using quantitative and qualitative methods. Before and after the application, a questionnaire form was applied to the children, and they used observations, interviews, and diaries as quantitative tools. Sullivan et al. (2017) also collected qualitative data with teachers' opinions and diaries in their study with pre-school children and used the Solve-Its measurement tool, which evaluates children's behaviors regarding robotic coding. Metin (2020) evaluated the training program he gave to 5-year-old children to support activity-based unplugged coding and robotic coding skills with the Basic Coding Skills Observation Form and the Robotic Coding Basic Skills Observation Form. Marinus et al. (2018) developed The Coding Development Test 3–6 (CODE Test 3–6) to measure cognitive skills that make up coding skills using robotic kits. In this context, it has become essential to develop a test to measure unplugged coding and robotic coding skills in early childhood based on activities.

Although observation and interview-based evaluations that measure children's coding skills provide information about the coding skills of the study groups, they do not provide appropriate tools for evaluating the studies conducted in larger groups. In addition, questionnaires and interviews include the subjective approach of the researchers, and they may be limited in expressing the thoughts of young children, especially in expressing the abstract process such as coding. Another limitation of the interviews and questionnaires used in previous studies is that these assessment tools are applied after coding training. Children already experience coding skills in their daily lives and are intertwined with digital technology. For this reason, the importance and need for assessment tools to reveal children's current coding skills and to determine children's current knowledge level, especially for educational programs to be prepared, is

increasing. There are many educational approaches and practices in the early years to support coding skills worldwide, and it is crucial to reveal the existing skills of children to demonstrate the adequacy of these applications and prepare programs for children's needs.

There is a strong tendency to increase coding education in the early childhood years to be based on experiences. For this reason, in the evaluation of coding skills of children of this age, it is seen that there is a need for assessment tools based on a positive technological development approach, requiring active participation of the child, and developmentally appropriate practices such as activities and games. For this reason, CodingTest, which measures coding and robotic coding skills, including concrete materials and stories, has been developed for early childhood (5–7 years old) children.

4 Method

4.1 Working group

The study group, which was designed to develop a scale for the coding and robotic coding skills of 5–7 years old children in early childhood, consists of children aged 5–7 who study in kindergarten, independent kindergarten, and primary schools affiliated with the Directorate of National Education in Gaziantep and Ağrı. Based on the opinions of the working group school administrators, the schools in the regions with lower, middle, and upper socioeconomic conditions were selected.

Fourteen schools, including seven primary schools, seven kindergartens in Gaziantep, 14 schools in total, eight primary schools, and six kindergartens in Ağrı, were included in the study. A total of 308 children participated in the study. The study included a total of 143 children, 69 girls and 74 boys from the 5–6–7 age group in Ağrı, were included in the study. Of these, 20 girls, 24 boys are at the lower socioeconomic level, 25 girls, 25 boys are at the middle socioeconomic level, 24 girls, 25 boys are at the upper socioeconomic level. A total of 165 children, 78 girls, and 87 boys, from the 5–6–7 age group from Gaziantep province, were included in the study. Of these, 25 girls, 27 boys are at the lower socioeconomic level, 26 girls, 30 boys are at the middle socioeconomic level, 27 girls, and 30 boys are at the upper socioeconomic level. The demographic information of the children participating in the study is given in Table 2.

Within the scope of the study, it was determined that 32.8% of the children were five years old, 32.5% were six years old, and 34.7% were seven years old, while 52.3% of the students were boys 47.7% were girls has been determined. It was determined that 35.39% of the children had not received a coding education before, while 20.78% received 0–1 year, 34.09% 1–2 years, and 9.74% 2–3 years.

4.2 Test development process

Generally, field experts (Conway, 2006) examine the test development process in five stages:

Table 2 Demographic information of the children of Ağrı and Gaziantep province participating in the study

Variable	Group	Ağrı	Gaziantep	Toplam	Yüzde (%)
		N	N		
Gender	Female	69	78	147	47,7
	Male	74	87	161	52,3
Age	5 years old	48	53	101	32,8
	6 years old	43	57	100	32,5
	7 years old	52	55	107	34,7
Coding education status	Not Trained	48	61	35,39	35,39
	0–1 Years	64	0	20,78	20,78
	1–2 Years	31	74	34,09	34,09
	2–3 Years	0	30	9,74	9,74
	3 Years and More	0	0	0,00	0,00

Defining the feature to be measured: It was decided by which questions to examine the learning products to be examined (Büyüköztürk, 2018). During this process, the literature was scanned, the studies on coding, online programs were examined, and the teachers' opinions who gave coding training were taken. CodingTest has been revealed by many researchers (Bers, 2012, 2018b; Bers et al., 2019); CSTA (2020) K - 12 Computer Science Standards were examined together with the opinions, and level 1A and ISTE (2016) standards for 5–7-year-old children were taken into consideration. The characteristics aiming to measure the unplugged coding and robotic coding skills that early childhood children should have (Table 1) were determined.

Item writing: After determining the relevant assessment tool's conceptual structure and main framework, the item pool related to the assessment tool was created. The items are prepared in a way to reveals the behavior to be observed. After the scale items were determined, Positive Technological Development (positive technological development) development and activity-based approach were used to observe the skills of the children regarding these scale items.

It provides a model for the Positive Technological Development (PTD) approach introduced by Bers (2010, 2012) to understand and examine how the use of technologies or technologically enriched contexts can support human development. This approach (PTD) considers the learning environment and pedagogical practices as well as cultural values and rituals that mediate teaching and learning (Bers, 2008, Bers, 2010; Rogoff et al., 2001; Horn et al., 2012; Bers et al., 2012).

Piaget (1964) and Bruner (1964) emphasized that toric knowledge is realized through activity, images, and symbolic reasoning, and abstract concepts can be gained through concrete denials. For this reason, Biazak et al. (2010), which enables children to learn by doing (Lijanporn & Khlaisang, 2015) and interact with the environment, activates children and stimulates children's curiosity and motivates them and ensures that this motivation continues throughout the process, encouraging problem-solving. (Batdi, 2014). Rodwell (2017) appeals to different

emotions and allows different learning styles; another feature is entertainment, the activity-based learning process that includes activities such as games, music, rhymes in the learning process, 3C (Curiosity, Creativity, Collaboration)) is based on curiosity, creativity cooperation. These activity-based features include scraping an abstract concept such as coding (Scharf et al., 2008; Futschek & Moschitz, 2010). Kamenetz (2016), McNerney (2004), Sullivan et al. (2015) formed the basis for observing the behavior of children regarding the items written, considering that coding would be a facilitator for children in the evaluation process.

Games and stories based on positive technological development and an activity-based approach were used for children to reveal the behaviors they wanted to measure. The plot in the stories is designed to measure a different feature of the child's coding. Pictures for each story and signs that the child can advance the coding process are included (Figs. 1 and 2). Care was taken to ensure that the visuals of these stories and stories are compatible with the age and developmental level of children, the structure and cultural characteristics of the society they live in and that they are prepared in the light of relevant literature and scientific data. Thus, the test booklet has been prepared. The measurement tool was prepared as A and B forms to measure coding skills in two structures: computerized coding and robotic coding skills.

Care was taken to ensure that the test items were clear, understandable, and fit for purpose, and instructions were prepared in this context. Checking whether it has the qualification to measure the behavior, whether there is a scientific mistake, developmental suitability, language comprehension, and whether the test and items are technically flawed (Baykul and Güzeller, 2015), and three computers in order to reveal the content validity. Moreover, instructional technology education, two child development, and education, two pre-school education, one assessment and evaluation, one graphic and visual arts, and one faculty member working in Turkish language and education. The test's application instruction and

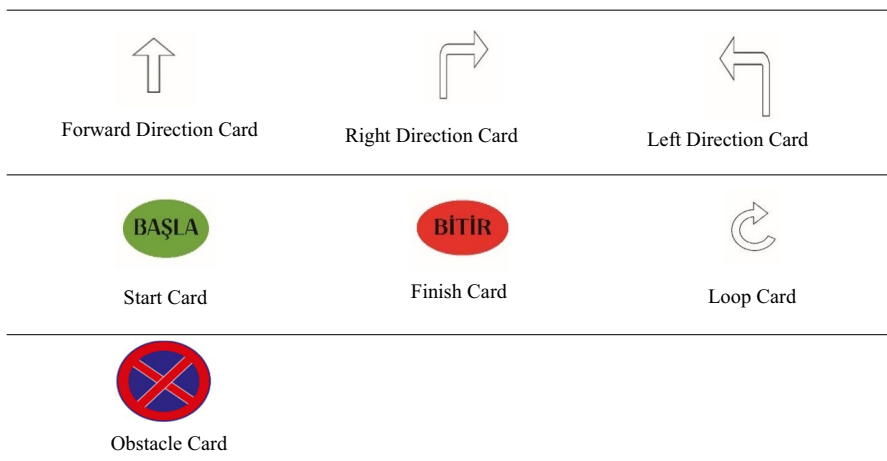


Fig. 1 Coding cards

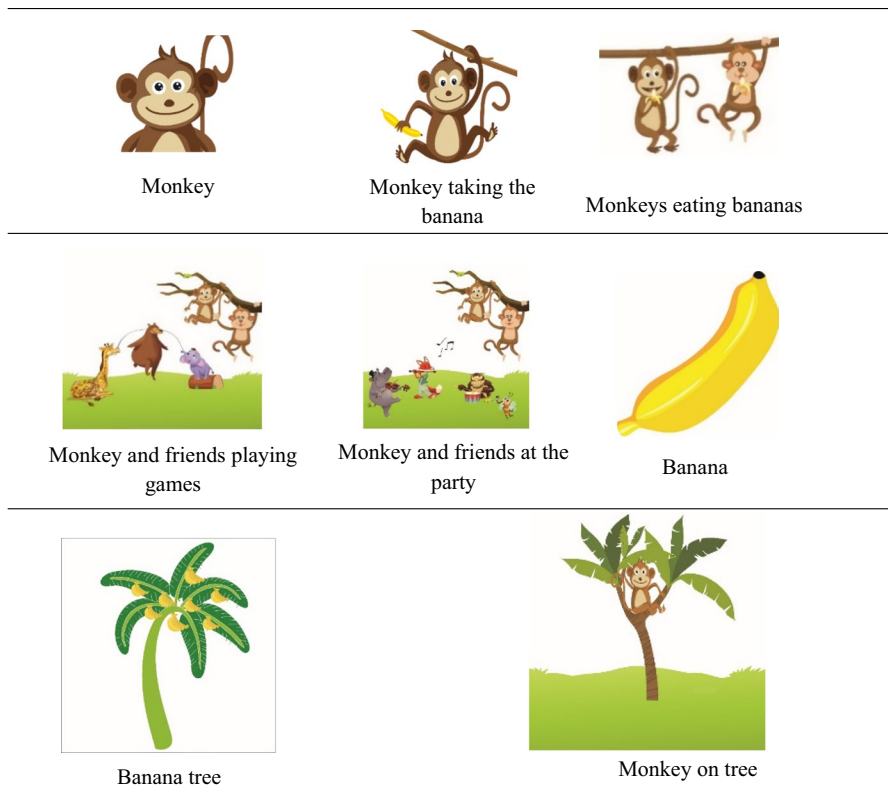


Fig. 2 Story cards

evaluation criteria have been arranged in line with the experts' recommendations and have been made more understandable and more straightforward in terms of grammar.

In addition, Lawshe Technique was used to evaluate expert opinions. Lawshe developed the Lawshe Technique, and for the application of the technique, a minimum of 5 and a maximum of 40 expert opinions are required (Lawshe, 1975). In the Lawshe Technique, the scope validity rates (CGO) are obtained by gathering expert opinions on the items (Yurdugül, 2005). In the form given to the experts, they were asked to mark the items as appropriate, not appropriate, and should be corrected, and to correct the items. In line with the data obtained from experts, the content validity of the items was statistically analyzed using the Lawshe technique. In this technique, ten experts' content validity rate of an item evaluated should not be below .99. Since the content validity rates of the items within the scale scope were not below .99, it was decided that there was no need to remove any item, and the analyses were finalized by making the necessary corrections in some items.

Writing the test form: CodingTest consists of two forms. CodingTest-A measures unplugged coding skills, CodingTest-B measures robotic coding skills. There

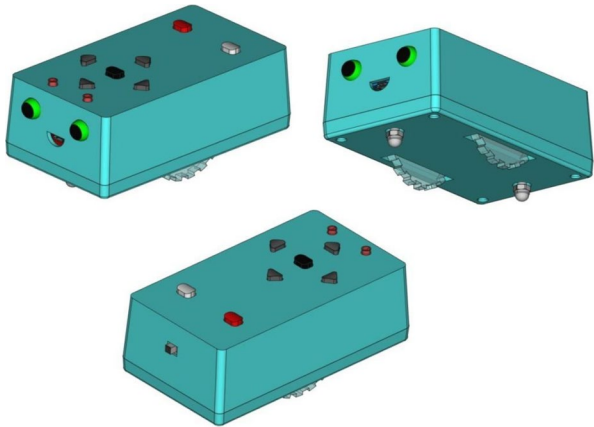
are 13 items in the CodingTest-A form and 14 items in the CodingTest-B form. In the test, each item is scored as “yes” (1 point), “no” (0 points). There is an explanation part in the form. The practitioner writes his comments on the application process in this section. The explanation part contributes to the practitioner’s interpretation of the results but does not affect the test score. CodingTest-A is the highest 13 points, the lowest 0 points, the highest 14 points, and the lowest 0 points from CodingTest-B. The children’s high scores from the test show that they have coding and robotic coding skills (Kalyenci, 2020).

There are eight stories in both tests. The first two stories of these eight stories are used for the sample application. Each story has a plot, and each measures one or more of the story coding skills. The stories are arranged in order to measure coding skills from easy to difficult. The practitioner reads the story, and the child is expected to encode the plot in this story on unplugged coding paper using story visuals and coding cards. While the child is coding, the practitioner observes the child and fills the application form. The practitioner first performs the sample application with the child. If the child makes the sample application, the actual application is started; the second sample application is made. The test is terminated with the child who cannot make the second sample application. The plot in practice 1 is to get the monkey to the banana. For this, the practitioner said to the child, *“We will play a game with you. Now listen carefully. This cute monkey dropped a banana while wandering in the forest. He searched for his banana in the depths of the forest and could not find it. The monkey is starving. The monkey will take the banana here and eat it. Can you show me the way the monkey will go first with your finger, then using the coding cards?”* it says and starts working. The story read by the practitioner is read clearly and understandably by making eye contact with the child (Kalyenci, 2020).

Test Materials: CodingTest-A Form includes unplugged coding paper, sign, and story cards. Unplugged coding paper is 9X9 square, and each square is 12X12 cm in size. It consists of pictures and sign (start-end cards, forward, right, left direction cards, obstacle, and loop) cards for the plot in each story. Story graphic cards and signal cards (Figs. 1 and 2) consist of 9X9 cm squares.

In the CodingTest-B form, there is robotic coding paper, cards with story images, and a robotic tool. “My School Bus” robotic kit tool was used as a robotic tool. The coding carpet consists of 6X6 squares. The size of each square is 15X15 cm. The story visuals are the same as the cards used in unplugged coding (Figs. 3 and 4).

Application: The test was written clearly and clearly to measure the coding and robotic coding skills of 5–7-year-old children. Before applying the test, a pre-application was made. The preliminary application was made with 45 children, 15 of them from every age group, taking into account different socioeconomic conditions (lower, middle, upper socioeconomic). The application was administered individually and lasted an average of 20–25 min for each child. During the implementation process, care was taken to ensure that the child and the researcher were present in a quiet environment and that the instructions in the application were read clearly and clearly. The pre-application results were examined, no problem was encountered regarding the understandability of the items, the applicability of the application steps, and the measurement tool was not changed (Kalyenci, 2020).

Fig. 3 Robotic coding paper**Fig. 4** My school bus robotic kit tool

5 Procedure

CodingTest A-B form was collected by the third researcher and three graduate students who were informed about the measurement tool by the researchers. Before practitioners started the practice, they had been with the classroom teacher and participated in classroom activities with the children for a while. Later, the children were taken one by one in a quiet room, and an individual application was made. First, the CodingTest-A form was applied, and after the CodingTest-A form was applied to the whole sampling, the Coding-B test was applied. While applying the CodingTest-A form, the practitioners lined up the practice cards together with the children on the table and chatted during this time. While applying the CodingTest-B form, the practitioners lined up the practice cards with the children on the table

together, chatted at this time, and allowed the children to examine the robotic kit for a while.

While applying the CodingTest-A-B form, the practitioners read the story of each application to the children and asked the children to show what they would do beforehand by drawing with their fingers and then using the coding cards. While the children were practicing, the practitioner filled the application form. After each story, the cards were put in place, and the following story was passed. The application took approximately 10–15 min.

Oğuz and Akyol for measuring children's problem-solving skills in early childhood for similar scale/compliance validity; The Problem Solving Skills Scale developed by (2015) was used. The scale includes problem situations to determine the problem-solving skills of children and a list of problem situations for these problem situations. There is a visual for each problem situation. By showing these images to the child, the problem situation sentence is read, and the child is expected to offer solutions to this problem situation. The scale scoring is 0 points if there is no solution suggestion, 1 point if there is only one suggestion, 2 points if there are two suggestions, 3 points if there are three suggestions, 4 points if there are more than three suggestions. As a result of the application of the scale, the total score obtained from each problem situation shows the child's problem-solving skills.

Coding skills also include problem-solving (Ackermann, 2001; Taylor et al., 2010; Wing, 2006; Wong et al., 2015; Csizmadia et al., 2015). For this reason, while evaluating children's coding skills, it was used as evidence for the similarity between coding and problem-solving skills.

6 Data analysis

In line with detailed information about the tests used in the study and the data obtained, item difficulty index, item discrimination index, and KR-20 reliability analysis were applied for the measurement tool's reliability within the study's scope.

For the validity analysis of the measurement tool developed; A versatile MANOVA test was conducted to see the effects of students' ages on their coding skill levels, an independent sample t-test was applied to see the differences in coding skill levels of students according to their previous coding training, the correlation between the measurement tool developed for similar scale validity and the Problem Solving Scale its value has been examined.

While the Cronbach Alpha internal consistency coefficient determines the reliability levels in the built scales, normally, the reliability levels are determined with the KR-20 test in the 2-tier structures in the form of Likert yes-no. According to the KR-20 formula, calculating is based on the assumption that each item in the test measures the same variable and has the same properties as the one measured by the test (Tekin, 2007). In order to analyze the reliability of the developed measurement tool, the KR-20 test was applied to 308 data. Reliability and validity statistics of the developed measurement tool are given in the findings section. Exploratory factor analysis based on tetrachoric correlation was preferred to provide the construct validity of the test developed within the scope of the study. Since the responses

given by the children to the items in the test were artificially transformed into two categories as 1–0, an exploratory factor analysis (EFA) was performed on the tetra-choric correlation matrix to determine the factor structures of the test. The tetra-choric correlation coefficient is used to determine the degree of relationship between two two-category artificial discontinuous variables (Baykul & Güzeller, 2014; Kan, 2011).

7 Findings

7.1 Validity analysis

For the construct validity to reveal the validity of the CodingTest, content validity (explained in the test development process), criterion validity, and construct-concept validity were examined.

7.1.1 Construct validity

For the construct validity, the pre-test study was applied to 45 children, and the comprehensibility of the instructions in the test was tested. In order to determine the test scores of the groups whose characteristics are known as the second step, a t-test was conducted to compare the scores of the children who had previously received coding training and those who did not. Construct validity explains the result obtained from the scale and what this result is related to. It is about how accurately the prepared scale items measure the determining properties (Akyüz, 2018). Results are presented in Table 3.

An independent sample t-test was conducted to determine the differences in the coding performances of children according to their previous coding training. According to the results of the independent sample t-test, it was determined that the difference between the coding performance of the children compared to the previous coding training status was significant ($t = 12.105$; $p = 0.000 < 0.01$). It was determined that the coding performance of the students who had received coding education before ($= 16.70$) was higher than the students who did not receive education before ($= 6.37$).

It was determined that the difference between the robotic coding performance of the students according to their previous coding training status was significant

Table 3 Independent sample T-Test result of the differences of CodingTest-A and CodingTest-B according to their coding education

Coding skills	Coding education status	n	\bar{x}	ss	t	p
CodingTest-A	No	109	6,37	5,70	-12,105	0,000**
	Yes	138	16,70	7,32		
CodingTest-B	No	109	8,36	7,08	-10,318	0,000**
	Yes	138	19,14	8,92		

Table 4 MANOVA analysis results of the effects of children’s ages on Coding Test-A and Coding Test-B

Source	Coding group	Type III sum of squares	sd	Average square	F	p	Partial Eta square
Age	CodingTest-A	547,461	2	273,730	3254	,040	,021
	CodingTest-B	2963,696	2	1481,848	12,983	,000	,078
Error	CodingTest-A	25,655,172	305	84,115			
	CodingTest-B	34,810,824	305	114,134			
Adjusted Total	CodingTest-A	26,202,633	307				
	CodingTest-B	37,774,519	307				

Table 5 Analysis results of the average of independent variables

Performance	Age	\bar{x}	Overall Avg.
CodingTest-A Form: Unplugged Coding Skills	5	12,86	14,75
	6	15,66	
	7	15,74	
CodingTest-B Form: Robotic Coding Skills	5	13,28	17,70
	6	19,52	
	7	20,18	

($t=10,318$; $p=0,000 < 0.01$). It was determined that the robotic coding performances of the students who had received coding education before ($= 19.14$) were at a higher level than the students who did not receive education before ($= 8.36$). This is the case for children who have received coding education before.

Construct validity, as another way, is based on the idea that children’s coding skills will differ by age (Büyüköztürk, 2018); in order to reveal the difference between groups with known characteristics, the MANOVA test was used to analyze whether the children differ according to their ages. Results are presented in Table 4.

Manova analysis was conducted to determine the effects of students’ ages on their coding performance. As a result of the MANOVA analysis, it was determined that there is a significant difference ($F=12.983$; $p=0.040$) in the coding performances of the age variable. When the corrected R square values were examined, it was determined that the age variable alone explains 1.4% of the coding performance. It was determined that the age variable has a significant effect ($p < 0.01$; $p=0.000$) on the performance of robotic coding. When the corrected R square values were examined, it was determined that the age variable alone explains 7.8% of the robotic coding performance.

When the averages in Table 5 are examined, it is seen that the average success in coding increases as the age increases, and it was determined that the average coding values are 14.75. In robotic coding, it is seen that as the age increases, the average success increases at a higher level than the coding, and it has been determined that the average success values of robotic coding are 17.70. These results show that the age variable is more effective on robotic coding than coding.

In addition, item difficulty index values were examined for the CodingTest-A and CodingTest-B forms. The item difficulty index shows the correct answer rate for each item in the test. The item difficulty index of an item takes a value between “0” and “1”. If this value is close to zero, it is interpreted that the substance is a problematic substance, and the fact that it is close to one is interpreted as an accessible substance. As a result of the item analysis, the item difficulty indexes of the items are between 0.20 and 0.80, and the average difficulty index of the test is around 0.50 (Tosun & Taşkesenligil, 2011). Form A: Coding Skills Test in Table 6 Form B: Item difficulty index results of the Robotic Coding Skills Test are given below in Table 6.

When the item difficulty index in the test was examined, it was determined that there was no need to remove items from the test according to the item difficulty index since the difficulty indexes of all items were above 0.20. When the item discrimination index in the test was examined, it was necessary to exclude item A.3.2 from the test in line with the expert opinions because the discrimination index value was below 0.19.

When the item difficulty index in the test was examined, it was determined that there was no need to remove items from the test according to the item difficulty index since the difficulty indexes of all items were above 0.20. When the item discrimination index in the test was examined, it was determined that there was no need to remove items from the test since the discriminatory indices of all items were over 0.19.

In order to determine the factor structures of the test, exploratory factor analysis (EFA) was performed on the tetrachoric correlation matrix. The tetrachoric correlation coefficient is used to determine the degree of relationship between two two-category artificial discontinuous variables (Baykul & Güzeller, 2014; Kan, 2011). The

Table 6 Item difficulty analysis results of the CodingTest-A/B form

Coding Test A		Coding Test B	
Item No	Item Difficulty Index	Item No	Item Difficulty Index
A1	0,57	B.1	0,78
A2	0,58	B.2	0,58
A3	0,73	B.3	0,86
A4	0,57	B.4	0,68
A.5	0,50	B.5	0,51
A.6	0,50	B.6	0,50
A.7	0,48	B.7	0,50
A.8	0,50	B.8	0,50
A.9	0,50	B.9	0,50
A.10	0,50	B.10	0,50
A.11	0,50	B.11	0,50
A.12	0,50	B.12	0,45
A.13	0,51	B.13	0,50
		B.14	0,53

reason for using EFA on the tetrachoric correlation matrix in this study is because the answers given by the students to the items in the test were artificially transformed into two categories as 1–0.

Exploratory factor analysis based on tetrachoric correlation was preferred to provide the construct validity of the test developed within the scope of the study. The analysis process was performed using the FACTOR program developed by Lorenzo-Seva and Ferrando (2013). The exploratory factor analysis is performed based on tetrachoric correlation because this technique is preferred in determining the correlation coefficient between items in measuring instruments scored as 0–1 (Olsson, 1979). The Hull method was preferred to determine the number of common factors to perform factor analysis. This is the finding that this method performs better than other methods used in determining the number of factors (Lorenzo-Seva et al., 2011). As the factor extraction method, minimum rank factor analysis (MRFA) was chosen. This is because the method allows seeing the variance ratios explained and unexplained (Socan, 2003). Thus the variance ratio explained by each factor can be interpreted (Lorenzo-Seva & Ferrando, 2006). After the factor number and factor extraction methods were decided, oblique rotation was chosen as the method of factor rotation in the program, technically Promin and factor loadings of .30 and above (Büyüköztürk, 2018) were preferred as the test items were not completely independent from each other. Within the scope of factor analysis performed under these conditions, the suitability of the data to factor analysis was first evaluated according to the Kaiser-Meyer-Olkin (KMO) coefficient and Barlett sphericity test results, and it was found that the KMO coefficient was .86 and the Barlett test was significant. In order to talk about the suitability of the data to factor analysis, the Barlett test with a KMO coefficient of over .60 should be significant (Tabachnick et al., 2007). Accordingly, it can be said that the data are suitable for factoring. As a result of the analysis performed, the program suggested that the test be one-dimensional. The variance rate explained in this way is 67.10%, and the opinion that the variance rate explained by a single factor should be at least 30% in the literature (Büyüköztürk, 2018), the opinion that the test can be one-dimensional has been strengthened.

7.1.2 Criterion-dependent validity

The validity technique that examines the relationship of test scores with one or more specified external criteria is called criterion-dependent validity (Büyüköztürk, 2018). For this purpose, the criterion-dependent validity of the CodingTest was determined by the randomly selected 100 children (53% female, 47% male), five years old 35 children (21 girls, 14 boys), six years old 32 children (13 girls, 19 boys), 7 Using the problem-solving skills scale developed by Oguz and Akyol (2015), 33 children (19 girls, 14 boys) and the problem-solving skills scale between the coding skills, the similar scale compatibility-relationship between the two scales was examined. The results of CodingTest and the “Problem Solving Scale” are presented in Table 7.

A relationship was found between the two scales at $r = .89$ and $r = .93$, $p < .01$ significance level in a similar scale validity study. This result was seen as another proof in determining the validity of CodingTest.

Table 7 Findings regarding the relationship between CodingTest and Problem solving scale

Variables		CodingTest-A	CodingTest-B	Problem Solving
CodingTest-A:	r	1	,891**	,838**
	p		,000	,000
CodingTest-B:	r		1	,928**
	p			,000
Problem Solving	r			1
	p			

**p < 0.01

7.2 Reliability analysis

7.2.1 Kuder-Richardson KR-20

The reliability analyzes of the CodingTest-A Form and the CodingTest-B test were examined.

When Table 8 is examined, KR-20 reliability analysis was performed to determine the internal consistency of the test and the reliability level of the CodingTest-A test ($KR-20 = 0.973 > 0.70$), and the CodingTest-B reliability level ($KR-20 = 0.978 > 0.70$) was determined to be high.

8 Discussion and future guidelines

Interest and educational applications for coding, which is seen as the language of our age, are spreading rapidly worldwide. In addition to being a new language of the age, coding skills also significantly affect thinking. While computer and unplugged applications are expected to support children's coding skills, there is no valid and reliable assessment tool to measure and evaluate this skill. For this reason, CodingTest was developed to evaluate the unplugged coding and robotic coding skills of 5–7-year-old children. CodingTest is designed as developmentally appropriate and activity-based based on the CSTA and ISTE standards and the skills revealed in the literature. The test's psychometric properties were tried to be revealed with 101 five-year-old, 100-six-year-old, 107-seven-year-old 308 children with different socioeconomic conditions.

Expert opinions of CodingTest: After the pre-application process, it has a structure to measure the unplugged coding and robotic coding skills of children between

Table 8 Form A: reliability analysis result regarding the coding skills test

Tests	KR-20	Number of Items
CodingTest-A	0,973	27
CodingTest-B	0,978	32

the ages of 5–7. In addition, to reveal the construct validity of the test, the coding skills of the children who had and did not receive coding training before were evaluated. It has been shown that there is a difference between the test scores of children who have received unplugged coding and robotic coding before, and this situation measures the determined skill of the measurement tool.

Structure validity is to determine the difference between known groups. For this reason, it was investigated whether there was a difference in the CodingTest scores of 5–7 year old children according to their ages. It was determined that the age variable had a significant effect on the CodingTest performances of the children. It was observed that as the age of the children increased, both unplugged coding and robotic coding success averages increased. In general, children's skills increase with age, developmentally. The increase in age from the CodingTest provided important evidence of the validity of the test.

Although the expert opinions on whether the items in the CodingTest measure the determining structure are proof, an exploratory factor analysis (EFA) was performed on the tetrachoric correlation matrix to determine the factor structures of the test. Within the scope of the factor analysis carried out, firstly, the suitability of the data for factor analysis was evaluated according to the Kaiser-Meyer-Olkin (KMO) coefficient and Barlett sphericity test results, and it was found that the KMO coefficient was .86 and the Barlett test was significant. According to the analysis results, it was seen that the data were suitable for factoring. As a result of the analysis performed, the program suggested that the test be one-dimensional. Following the purpose of the study, CodingTest was applied to students in schools located in regions with lower, middle, and upper socioeconomic conditions, taking into account the different socioeconomic conditions of the schools affiliated to the Ministry of National Education, in order to determine different situations and gave the same result of the scale in groups with different socioeconomic conditions. The total score is obtained from the double Likert type measurement tool, and a high score indicates that children's coding skills are high. The A form of the test measures unplugging children's coding skills, and the B form measures their robotic coding skills. In order to evaluate coding skills, the A and B forms of the test can be used together or separately. The test gives the children the flexibility to decide which form to use according to the type of education or practices given or the preference of researchers or practitioners.

Overall, CodingTest has been demonstrated to be a valid and reliable tool for measuring unplugged coding and robotic coding skills in children aged 5–7 years. In previous studies on coding and robotic coding, evaluation methods such as observation, questionnaire, interview, test (Bers, 2019; Sullivan & Bers, 2016; Metin, 2020; Marinus et al., 2018) were used. In the development process of CodingTest, all these measurement tools guided and shed light on the development of a valid and reliable measurement tool.

CodingTest is a valid and reliable tool to measure the skills of unplugged coding and robotic coding of children aged 5–7 and provide information on the unplugged coding and robotic coding skills that should be acquired by children aged 5–7. This measurement tool will also contribute to the educators' awareness of coding skills and their evaluation of the educational content in the pieces of training given to their children. Early childhood is a period that requires developmentally different

teaching methods and techniques for children. For this reason, the educational experiences given in this period are carried out through activities such as games, stories, and music. Although children experience coding skills in their daily lives, Lee & Junoh, (2019); Resnick & Siegel, (2015); Sullivan & Bers, (2016); Ananiadou & Claro, (2009), educational practices for this skill mainly involve the use of computer-based or puzzle-type approaches as Bers et al. (2019) stated, and are insufficient to support coding skills. Considering the feature of this period, the application process of CodingTest also consists of stories containing a plot. The developmental compatibility approach leading to the development of the test and the fact that it includes activity-based applications will also guide the preparation of the training content to be given on this subject.

According to the results obtained from this study; It is planned to standardize the test, reveal its normative values, and revise the scale in a way that does not require an application to children. It is thought that a tool can be developed to measure the coding skills of children aged 5–7 in the computer environment. It is suggested that it can be developed in older age groups, considering the coding skills revealed by CSTA and ISTE.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval All procedures performed in studies involving human participants were following the ethical standards of the Hasan Kalyoncu University IRB protocol no 804.01.

Informed consent Informed consent was obtained from the educators and parents/guardians of participating students. The students gave their assent for inclusion.

References

- Ackermann, E. (2001). Piaget's constructivism, Papert's constructionism: What's the difference. *Future of learning group publication*, 5(3), 438.
- Akyüz, H. E. (2018). Yapı geçerliliği için doğrulayıcı faktör analizi: Uygulamalı bir çalışma. *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*, 7(2), 186–198. <https://doi.org/10.17798/bitlisfen.414490>
- Ananiadou, K., & Claro, M. (2009). 21st century skills and competences for new millennium learners in OECD countries. OECD education working papers, no. 41. *OECD Publishing*. <https://doi.org/10.4018/978-1-60566-274-9.ch001>.
- Arabacıoğlu, T., Bülbül, H. İ., & Filiz, A. (2007). Bilgisayar programlama öğretiminde yeni bir yaklaşım. *Akademik bilişim*. <http://hdl.handle.net/11607/571>
- Batdi, V. (2014). The effects of a problem based learning approach on students' attitude levels: A meta-analysis. *Educational Research and Reviews*, 9(9), 272. <https://doi.org/10.5897/ERR2014.1771>
- Bateman, A., Carr, M., & Gunn, A. C. (2017). Children's use of objects in their storytelling. *Interactions in Early Childhood Education*, 37-53. https://doi.org/10.1007/978-981-10-4879-1_4.
- Baykul, Y., & Güzeller, C. O. (2014). *Sosyal bilimler için istatistik: SPSS uygulamalı*. Pegem Akademi Yayıncılık.
- Bell, T., & Vahrenhold, J. (2018). CS unplugged—How is it used, and does it work? *In Adventures between lower bounds and higher altitudes* (pp. 497-521). Springer. https://doi.org/10.1007/978-3-319-98355-4_29.

- Bell, T., Rosamond, F., & Casey, N. (2012). Computer science unplugged and related projects in math and computer science popularization. In *The multivariate algorithmic revolution and beyond* (pp. 398–456). Springer. https://doi.org/10.1007/978-3-642-30891-8_18.
- Bers, M. U. (2010). The TangibleK robotics program: Applied computational thinking for young children. *Early Childhood Research Practice, 12*(2), n2 <http://ecrp.uiuc.edu>
- Bers, M. U. (2012). *Designing digital experiences for positive youth development: From playpen to playground*. Oxford University Press.
- Bers, M. U. (2018a). Coding and computational thinking in early childhood: The impact of ScratchJr in Europe. *European Journal of STEM Education, 3*(3), 8. <https://doi.org/10.20897/ejsteme/3868>.
- Bers, M. U. (2018b). Coding, playgrounds and literacy in early childhood education: The development of KIBO robotics and ScratchJr. *2018 IEEE global engineering education conference (EDUCON)*, 2094–2102. <https://doi.org/10.1109/EDUCON.2018.8363498>.
- Bers, M. U. (2019). Coding as another language: A pedagogical approach for teaching computer science in early childhood. *Journal of Computers in Education, 6*(4), 499–528. <https://doi.org/10.1007/s40692-019-00147-3>
- Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers Education, 72*, 145–157. <https://doi.org/10.1016/j.compedu.2013.10.020>
- Bers, M. U., González-González, C., & Armas-Torres, M. B. (2019). Coding as a playground: Promoting positive learning experiences in childhood classrooms. *Computers Education, 138*, 130–145. <https://doi.org/10.1016/j.compedu.2019.04.013>
- Biazak, J. E., Marley, S. C., & Levin, J. R. (2010). Does an activity-based learning strategy improve preschool children's memory for narrative passages? *Early Childhood Research Quarterly, 25*(4), 515–526. <https://doi.org/10.1016/j.ecresq.2010.03.006>
- Briggs, J. D. (2013). *Programming with scratch software: The benefits for year six learners* [Unpublished master thesis]. Bath Spa University.
- Bruner, J. S. (1964). The course of cognitive growth. *American Psychologist, 19*(1), 1–15. <https://doi.org/10.1037/h0044160>
- Büyüköztürk, Ş. (2018). *Sosyal bilimler için veri analizi el kitabı*. Nobel Yayıncılık.
- Campbell, C., & Walsh, C. (2017). Introducing the 'new' digital literacy of coding in the early years. *Practical Literacy: The Early Primary Years, 22*(3), 10–12.
- Cejka, E., Rogers, C., & Portsmouth, M. (2006). Kindergarten robotics: Using robotics to motivate math, science, and engineering literacy in elementary school. *International Journal of Engineering Education, 22*(4), 711.
- Conway, M. J. (2006). *How to collect data: Measurement & Evaluation*. American Society for Training and Development.
- CSTA, K-12. (2003). Computer science standards. *Computer Science Teachers Association, 12*.
- CSTA, K-12. (2011). Computer science standards. *Computer Science Teachers Association, 12*.
- CSTA, K-12. (2019). Computer science standards. *Computer Science Teachers Association, 12*.
- CSTA, K-12. (2020). Computer science standards. *Computer Science Teachers Association, 12*.
- Demirer, V., & Sak, N. (2016). Programming education and new approaches around the world and in Turkey/Dünyada ve Türkiye'de programlama eğitimi ve yeni yaklaşımlar. *Eğitimde Kuram ve Uygulama, 12*(3), 521–546 <http://acikerisim.lib.comu.edu.tr:8080/handle/COMU/1443>
- Fesakis, G., & Serafeim, K. (2009). Influence of the familiarization with "scratch" on future teachers' opinions and attitudes about programming and ICT in education. *Acm SIGCSE Bulletin, 41*(3), 258–262. <https://doi.org/10.1145/1595496.1562957>.
- Flannery, L. P., & Bers, M. U. (2013). Let's dance the "robot hokey-pokey!" children's programming approaches and achievement throughout early cognitive development. *Journal of Research on Technology in Education, 46*(1), 81–101. <https://doi.org/10.1080/15391523.2013.10782614>
- Fleer, M. (2013). Affective imagination in science education: Determining the emotional nature of scientific and technological learning of young children. *Research in Science Education, 43*(5), 2085–2106. <https://doi.org/10.1007/s11165-012-9344-8>
- Futschek, G. (2006). Algorithmic thinking: The key for understanding computer science. *International conference on informatics in secondary schools-evolution and perspectives*, 159–168. https://doi.org/10.1007/11915355_15.
- Futschek, G., & Moschitz, J. (2010). Developing algorithmic thinking by inventing and playing algorithms. *Proceedings of the constructionist approaches to creative learning, thinking education: Lessons for the 21st century*, 1–10.

- Futschek, G., & Moschitz, J. (2011). Learning algorithmic thinking with tangible objects eases transition to computer programming. *International conference on informatics in schools: Situation, evolution, and perspectives*, 155–164. https://doi.org/10.1007/978-3-642-24722-4_14.
- García Peñalvo, F. J., Reimann, D., Tuul, M., Rees, A., & Jormanainen, I. (2016). An overview of the most relevant literature on coding and computational thinking with emphasis on the relevant issues for teachers. <https://doi.org/10.5281/zenodo.165123>.
- Gibson, J. P. (2012). Teaching graph algorithms to children of all ages. *Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education*, 34–39. <https://doi.org/10.1145/2325296.2325308>
- Gujberova, M., & Kalas, I. (2013). Designing productive gradations of tasks in primary programming education. In *Proceedings of the 8th workshop in primary and secondary computing education* (pp. 108–117). <https://doi.org/10.1145/2532748.2532750>.
- Horn, M. S., & Jacob, R. J. (2007). Designing tangible programming languages for classroom use. *Proceedings of the 1st international conference on tangible and embedded interaction*, 159–162. <https://doi.org/10.1145/1226969.1227003>.
- Horn, M. S., Crouser, R. J., & Bers, M. U. (2012). Tangible interaction and learning: The case for a hybrid approach. *Personal Ubiquitous Computing*, 16(4), 379–389. <https://doi.org/10.1007/s00779-011-0404-2>
- ISTE (2016). ISTE Standards For Students (Permitted Educational Use). <https://www.iste.org/standards>
- ISTE (2019). *International Society for Technology in Education* [Conference presentation]. Philadelphia. <https://www.conference.iste.org/2019>
- Johnston, K., Highfield, K., & Hadley, F. (2018). Supporting young children as digital citizens: The importance of shared understandings of technology to support integration in play-based learning. *British Journal of Educational Technology*, 49(5), 896–910. <https://doi.org/10.1111/bjet.12664>
- Kalyenci, D. (2020). *Erken çocukluk döneminde kodlama becerilerinin değerlendirilmesi-Test geliştirme [Yayınlanmamış yüksek lisans tezi]*. Hasan Kalyoncu Üniversitesi.
- Kamenetz, A. (2016). Half of professors in NPR Ed survey have used “trigger warnings.”. *National Public Radio*.
- Kan, A. (2011). Test eşitleme: OKS testlerinin istatistiksel eşitliğinin sınanması. *Eğitim ve Bilim*, 36(160).
- Kazakoff, E. R., Sullivan, A., & Bers, M. U. (2013). The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. *Early Childhood Education Journal*, 41(4), 245–255. <https://doi.org/10.1007/s10643-012-0554-5>
- Lawshe, C. H. (1975). A quantitative approach to content validity. *Personnel Psychology*, 28(4), 563–575.
- Lee, F., & Björklund Larsen, L. (2019). How should we theorize algorithms? *Five ideal types in analyzing algorithmic normativities*. <https://doi.org/10.1177/2053951719867349>
- Lee, J., & Junoh, J. (2019). Implementing unplugged coding activities in early childhood classrooms. *Early Childhood Education Journal*, 47(6), 709–716. <https://doi.org/10.1007/s10643-019-00967-z>
- Lee, K. T., Sullivan, A., & Bers, M. U. (2013). Collaboration by design: Using robotics to foster social interaction in kindergarten. *Computers in the Schools*, 30(3), 271–281. <https://doi.org/10.1080/07380569.2013.805676>
- Levy, S. T., & Mioduser, D. (2008). Does it “want” or “was it programmed to...”? Kindergarten children’s explanations of an autonomous robot’s adaptive functioning. *International Journal of Technology Design Education*, 18(4), 337–359. <https://doi.org/10.1007/s10798-007-9032-6>
- Lijanporn, S., & Khlaisang, J. (2015). The development of an activity-based learning model using educational mobile application to enhance discipline of elementary school students. *Procedia-Social Behavioral Sciences*, 174, 1707–1712. <https://doi.org/10.1016/j.sbspro.2015.01.825>
- Lorenzo-Seva, U., & Ferrando, P. J. (2006). Factor: A computer program to fit the exploratory factor analysis model. *Behavior Research Methods*, 38(1), 88–91. <https://doi.org/10.3758/BF03192753>
- Lorenzo-Seva, U., & Ferrando, P. J. (2013). Factor 9.2: A comprehensive program for fitting exploratory and semiconfirmatory factor analysis and IRT models. *Applied Psychological Measurement*. <https://doi.org/10.1177/0146621613487794>.
- Lorenzo-Seva, U., Timmerman, M. E., & Kiers, H. A. (2011). The Hull method for selecting the number of common factors. *Multivariate Behavioral Research*, 46(2), 340–364. <https://doi.org/10.1080/00273171.2011.564527>
- Manches, A., & Plowman, L. (2017). Computing education in children’s early years: A call for debate. *British Journal of Educational Technology*, 48(1), 191–201. <https://doi.org/10.1111/bjet.12355>

- Marinus, E., Powell, Z., Thornton, R., McArthur, G., & Crain, S. (2018). 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*, 133–141. <https://doi.org/10.1145/3230977.3230984>.
- McLennan, D. P. (2017). Creating coding stories and games. *Teaching Young Children*, 10(3), 18–21.
- McLennan, D. P. (2018). Code breaker: Increase creativity, remix assessment, and develop a class of coder ninjas! *Journal of Teaching Learning*, 12(1), 51–52 <http://www.uwindsor.ca/jtl>
- McNerney, T. S. (2004). From turtles to tangible programming bricks: Explorations in physical language design. *Personal Ubiquitous Computing*, 8(5), 326–337. <https://doi.org/10.1007/s00779-004-0295-6>
- Metin, S. (2020). Activity-based unplugged coding during the preschool period. *International Journal of Technology Design Education*, 1–17. <https://doi.org/10.1007/s10798-020-09616-8>.
- Mittermeir, R. T. (2013). Algorithmics for preschoolers—A contradiction? *Creative Education*, 4(09), 557. <https://doi.org/10.4236/ce.2013.4908>
- Montemayor, J. (2003). Physical programming: tools for kindergarten children to author physical interactive environments. <http://www.cs.umd.edu/~monte/papers/dissertation.pdf>
- Oguz, V., & Akyol, A. K. (2015). Problem Çözme Becerisi Ölçeği (PÇBÖ) Geçerlik ve Güvenirlilik Çalışması. *Çukurova University. Faculty of Education Journal*, 44(1), 105. <https://doi.org/10.14812/cufej.2015.006>.
- Olsson, U. (1979). Maximum likelihood estimation of the polychoric correlation coefficient. *Psychometrika*, 44(4), 443–460. <https://doi.org/10.1007/BF02296207>
- P21 Partnership for 21st Century Learning. (2017). 21st century learning for early learning: Framework. <http://www.p21.org/component/content/article/1-home/2233-21st-century-learning-for-early-childhood>
- Papert, S. (1980). *Teaching children to be mathematicians us. Teaching about mathematics*.
- Piaget, J. (1964). Cognitive development in children: Piaget. *Journal of Research in Science Teaching*, 2(3), 176–186.
- Piaget, J. (1970). *Science of education and the psychology of the child*. Trans. D. Coltman.
- Reese, E., Gunn, A., Bateman, A., & Carr, M. (2019). Teacher-child talk about learning stories in New Zealand: A strategy for eliciting children’s complex language. *Early Years*, 1–16. <https://doi.org/10.1080/09575146.2019.1621804>
- 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 Technology*, 29, 482–498. <https://doi.org/10.1007/s10956-020-09831-x>
- Resnick, M., & Siegel, D. (2015). A different approach to coding. *International Journal of People-Oriented Programming*, 4(1), 1–4. <https://doi.org/10.4018/IJPOP.2015010101>
- Rogoff, B., Baker-Sennett, J., Lacasa, P., & Goldsmith, D. (1995). Development through participation in sociocultural activity. *New Directions for Child Adolescent Development*, 1995(67), 45–65. <https://doi.org/10.1002/cd.23219956707>
- Rogoff, B., Turkanis, C. G., & Bartlett, L. (2001). *Learning together: Children and adults in a school community*. Oxford University Press.
- Scharf, F., Winkler, T., & Herczeg, M. (2008). Tangicons: Algorithmic reasoning in a collaborative game for children in kindergarten and first class. *Proceedings of the 7th international conference on interaction design and children*, 242–249. <https://doi.org/10.1145/1463689.1463762>.
- Socan, G. (2003). *The incremental value of minimum rank factor analysis*. University of Groningen.
- Stoeckelmayr, K., Tesar, M., & Hofmann, A. (2011). Kindergarten children programming robots: A first attempt. *Proceedings of 2nd International Conference on Robotics in Education (RIE)*.
- Sullivan, A., & Bers, M. U. (2016). Robotics in the early childhood classroom: Learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade. *International Journal of Technology Design Education*, 26(1), 3–20. <https://doi.org/10.1007/s10798-015-9304-5>
- Sullivan, A., Kazakoff, E. R., & Bers, M. U. (2013). The wheels on the bot go round and round: Robotics curriculum in pre-kindergarten. *Journal of Information Technology Education*, 12, 203–219 <http://www.jite.org/documents/Vol12/JITEv12IIPp203-219Sullivan1257.pdf>
- Sullivan, A., Elkin, M., & Bers, M. U. (2015, June). KIBO robot demo: Engaging young children in programming and engineering. In *Proceedings of the 14th international conference on interaction design and children* (pp. 418–421). <https://doi.org/10.1145/2771839.2771868>.

- Sullivan, A. A., Bers, M. U., & Mihm, C. (2017). Imagining, playing, and coding with KIBO: Using robotics to foster computational thinking in young children. *Siu-cheung KONG the Education University of Hong Kong, Hong Kong*, 110.
- Tabachnick, B. G., Fidell, L. S., & Ullman, J. B. (2007). *Using multivariate statistics* (Vol. 5, pp. 481–498). Boston, MA: Pearson.
- Taylor, M., Harlow, A., & Forret, M. (2010). Using a computer programming environment and an interactive whiteboard to investigate some mathematical thinking. *Procedia-Social and Behavioral Sciences*, 8, 561–570. <https://doi.org/10.1016/j.sbspro.2010.12.078>
- Tosun, C., & Taşkesenligil, Y. (2011). Revize edilmiş Bloom'un taksonomisine göre çözümler ve fiziksel özellikleri konusunda başarı testinin geliştirilmesi: Geçerlik ve güvenilirlik çalışması. *Kastamonu Eğitim Dergisi*, 19(2), 499–522.
- Vorderman, C. (2019). *Computer coding for kids: A unique step-by-step visual guide, from binary code to building games*. Dorling Kindersley Ltd..
- Vygotsky, L. S. (1980). *Mind in society: The development of higher psychological processes*. Harvard university press.
- Wang, D., Zhang, C., & Wang, H. (2011). T-Maze: A tangible programming tool for children. *Proceedings of the 10th international conference on interaction design and children*, 127-135. <https://doi.org/10.1145/1999030.1999045>.
- Welch, L., Kozlowski, J., & Evans, H. (2019). Coding to develop early mathematical and computational thinking in kindergarten: A case study.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35. <https://doi.org/10.1145/1118178.1118215>
- Wong, K. W. G., Ching, C. C., Mark, K. P., Tang, J. K., Lei, C. U., Cheung, H. Y., & Chui, H. L. (2015). Impact of computational thinking through coding in K-12 education: A pilot study in Hong Kong. *General Studies*, 85(88.01), 2-08.
- Yurdugül, H. (2005). Ölçek geliştirme çalışmalarında kapsam geçerliği için kapsam geçerlik indekslerinin kullanılması. *XIV. Ulusal Eğitim Bilimleri Kongresi*, 1, 771–774.
- Zamin, N., Ab Rahim, H., Savita, K., Bhattacharyya, E., Zaffar, M., & Jamil, S. N. K. M. (2018). Learning block programming using scratch among school children in Malaysia and Australia: An exploratory study. In *2018 4th international conference on computer and information sciences (ICCOINS)* (pp. 1–6). <https://doi.org/10.1109/ICCOINS.2018.8510586>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.