Understanding Problem-Solving Processes of Preschool Children in CS- Unplugged Activities

Ünal ÇAKIROĞLU[1]
Şüheda MUMCU[1]
Melek ATABAY[1]
Merve AYDIN[1]

[1]Trabzon University

DOI: 10.21585/ijcses.v5i3.133

Abstract

This study aims to explore the influences of the CS-unplugged activities in developing problem solving skills of preschool children. The participants were 11 children (4-5 aged) enrolled in a public preschool and Code.org activities were used as an instructional package. Activity evaluation form and interviews were used to understand children's problem solving processes. In order to determine the problem solving performances, the tasks were divided into the meaningful sub-tasks with regard to problem steps of Nance' problem solving model. The results indicated that CS-unplugged activities positively influenced students' understanding and planning performances more than doing and evaluation skills. Preschool children developmental characteristics and the nature of the problems somewhat hampered the development of their performances in doing and evaluation steps. It is hoped that the study may provide insights for the efforts on enhancing preschool children's problem solving processes.

Key words: CS unplugged activities; problem solving; preschool children

1. Introduction

Over the past decade, computational thinking (CT) has become a very hot topic in educational research and practice. After Wing's (2006) declaration, a common idea for CT definition entails at least thinking in a way that formulating problems and their solutions are represented in a form that can be effectively carried out with an information-processing agent (Wing, 2011). It leaded researchers to study on supporting young children to acquire thinking skills that are transferable to problem solving in computing related subjects (Bransford et al., 2000, Brackmann et al., 2017).

Wing (2006) argued that CT is a fundamental skill for everyone, not just for computer scientists. Researchers also highlighted that CT is an important skill that should be taught to the next generation (Barr et al., 2011; Brown et al., 2013; Grgurina et al., 2014; Grover & Pea, 2013; Hodhod et al., 2016; Kafai & Burke, 2013; Voogt et al., 2015; Wing, 2006). Thus, many countries have updated their computer science (CS) curriculum to teach children starting from young ages (Bargury et al., 2012; Bers et al., 2014; Grgurina et al., 2014; Grout & Houlden, 2014; Kalelioglu et al., 2014; Lee, Martin, & Apone, 2014; Repenning, Webb, & Ioannidou, 2010). In an attempt to increase interest in CS, much effort has gone into developing some preliminary learning materials, activities, methods or tools for teaching CT for young children. In these studies, programming (Dann, Cooper, & Pausch, 2009; Resnick et al., 2009), educational robotics and CS-unplugged activities (Bell, Witten, & Fellows, 2005, Wohl et al., 2015) are frequently used considering the educational levels.

Since it is difficult to teach CT to young children via programming or robotics, CS-unplugged activities are suggested as an introduction strategy (Battal, Adanır & Gülbahar, 2021). Additionally, not all children are lucky enough to access to powerful tools and toys and CS-unplugged activities yield equal opportunity across all learners as individuals that do not require so many technological tools. Unplugged activities also have potentials to learn the concepts without the need for technological devices or computers (Kalelioğlu & Keskinkılıç, 2017).

Previous research has demonstrated beneficial effects of CS-unplugged initiative as a way of teaching CT in early ages. For example, Del Olmo-Munoz ˜ et al. (2020) found that in the early stages of primary education, it is more suitable to perform CT through unplugged programming activities before plugged-in activities. Bell and Vahrenhold (2018) found through a literature review that unplugged programming activities can help students and teachers stimulate motivation to explore CS in a meaningful and attractive way and can also help students to carry out subsequent 'plugged-in' learning. Wohl et al. (2015) compared Scratch, Cubelets, and unplugged activities in teaching CS to 5-7 aged children and found that unplugged activities are more powerful in teaching concept of algorithm than others. On the other hand, Caelien and Yadav (2020) pointed out that unplugged programming activities can support students' participation in plugged-in programming activities in the future. Some other researchers also emphasized the role of CS-unplugged activities as a priming step to help students understand algorithmic steps before they write code (Gardeli &Vosinakis, 2017; Uchida et al., 2015). Following conclusions from the previous studies we aim at gaining an insight into the relationships between the nature of CS-unplugged activities and the problem-solving process of preschool students.

1.1. CS-unplugged Activities for Developing Problem Solving Skills

According to Wing (2008) CS-unplugged activities are various kinds of problems that do not directly involve coding tasks. CS-unplugged is defined as a widely used collection of activities and ideas to engage a variety of audiences with great ideas from CS, without having to learn programming or even use a digital device (Bell & Vahrenhold, 2018, p. 497). Research have shown that CS-unplugged activities contribute to the acquisition of basic CS concepts (Hermans & Aivaloglou 2017; Wohl et al. 2015; Taub et al., 2009), support improvement of CT (Leifheit et al., 2018; Jagušt et al., 2018; Rodriguez, 2015), provide entertainment for the lesson (Bell & Vahrenhold 2018; Curzon, 2014) and help to overcome misconceptions or negative attitudes towards programming (Bell & Vahrenhold 2018). Researchers argued that using CS-unplugged activities would break the wall between CS and using computers in real life problem solving for children of young ages (Nishida et al., 2008; Lambert and Guiffre, 2009; Bell & Vahrenhold, 2018). Ahn, Sung, and Black (2021) also reported that CS-unplugged activities enhance younger students' problem-solving skills, debugging, and confidence, and to reduce the obstacles that programming can present for novice learners. Besides, unplugged approaches may be less intimidating to teachers without a background in CS or programming and avoid the high costs of teaching coding or dealing with hardware (Huang & Looi, 2021). Taking their advantages in learning with games, trial-and-error with real objects and learning within groups have made CS-unplugged popular in problem solving activities (Nishida et al., 2008).

There is a significant research effort invested on discussing the effects of CS-unplugged activities on problem solving skills to convey fundamental CS concepts to children without any computer skills in the schools (Bell et al., 2009; Prottsman, 2014; Wohl et al., 2015). According to Dwyer et al. (2014), while acting in CS-unplugged activities young children can describe problems, identify the requisites to solve the problem, break the problem into small logical steps, use these steps to meaningful problem-solving process, and then evaluate this process. In this sense, Alamer et al. (2015) used CS-unplugged in a camp to introduce programming concepts. In another study, Dwyer et al. (2014) implemented CS-unplugged to measure students' ability to work with systematic instructions in algorithms. In secondary education, Thies and Vahrenhold (2012) used CS-unplugged activities and addressed positive results in CT skills of students. There have been some projects undertaken to propose the potentials of CS-unplugged. For instance, Bebras is a test of computational problem solving that does not require the use of a programming language (Dagienė, et al., 2016; Gujberova & Kalas, 2013). In addition, (csunplugged.org) by CS Education Research Group in New Zealand introduces CT principles without using a computer (Bell, Witten, & Fellows, 2015). Another popular CS-unplugged project is Code.org. It introduces a blocky coding platform for preschool students through the 8th grade and older. It also covers a variety of algorithmic concepts that are connected to everyday life dedicated for children from the 4th year.

In order to improve problem-solving skills of young children, researchers suggested preparing activities focusing on children's developmental characteristics (Çetin, 2016). Although an increasing number of nations have plans for introducing CS-unplugged activities in early childhood, problem solving activities within CS-unplugged activities are not formally integrated into the preschool curriculum. Thus, a need exists to present models to guide the educators. Following the idea that CS-unplugged activities can promote young children to engage better in problem solving activities; this study seeks to examine the influences of CS-unplugged activities on their problem-solving skills.

1.2. Research Problem

The aim of this research is exploring the development of problem-solving skills of preschool children with CS-unplugged activities. More specifically, the research question is "To what extent do the CS activities-unplugged develop young children's problem-solving skills?" was investigated.

2. Method

In this study, an instructional package including CS-unplugged activities was used for developing problem solving skills was implemented. A sequential explanatory mixed method was implemented in this research. This study undertakes the sequential approach where the quantitative phase is followed by the qualitative phase and the qualitative findings are used to contextualize the quantitative ones (Creswell, et al., 2003). Activity evaluation forms were used for evaluating the problem-solving skills quantitatively and interview data were used to explain the reasons of developments in problem solving skills through the students' experiences.

During implementation, one of the researchers was observed the students' behaviors in their learning environment as a participant researcher and tried to understand the atmosphere, language or views of the group. At the same time, observation data were also used in the analysis of qualitative data by observing the group in depth through this participated researcher.

2.1. Participants

This study was carried out with 11 children (aged 48-60 months) enrolled in a preschool. They did not take similar activities focusing directly problem-solving skills. They normally show the basic developmental characteristics behaviors of their young age during implementations which took place in the class environment.

2.2. Process

The study lasted 5 weeks, 2 class hours per week with Code.org unplugged activities. Children at the age of 48-72 months can perform activities such as matching, establishing cause-effect relations, reading object graph and creating graphics with regard to their developmental characteristics (Piaget, 1976). Accordingly, the activities in Code.org were selected considering the motor, linguistics, cognitive and social development of young children characteristics. The activities covered direction, rhythm and classification skills including loops, conditionals or patterns. The children can provide different decisions, learn to carry out the iterative process to achieve tasks and produce tangible artifacts. The acquisitions covered in the activities are presented in Appendix1. Three experts (2 preschool and 1 IT experts) reviewed the activities for content validity regarding the covered skills. Activities were selected based on in the 2013 preschool education program of the MoNE for 36-72 months old children considering the grouping and thinking skills on Code.org (Table 1). All activities in the Code.org have developmental foundations and evaluation worksheets, as well as various daily life problems appropriate for all age groups.

Code.org worksheet assessment forms were followed to develop a detailed lesson plan that would be applied in a 30-min. class period. The teacher introduced materials and basic problem-solving activities by following the lesson plans. The children worked on the tasks around common tables and followed the worksheets to complete the activity individually. The researcher only guided the children when they did not understand the tasks but did not explain how to solve the problem. The researcher as an observer took notes by observing the children's behaviors and filled the evaluation form regarding their problem-solving performances. The research process is summarized in Figure 1.

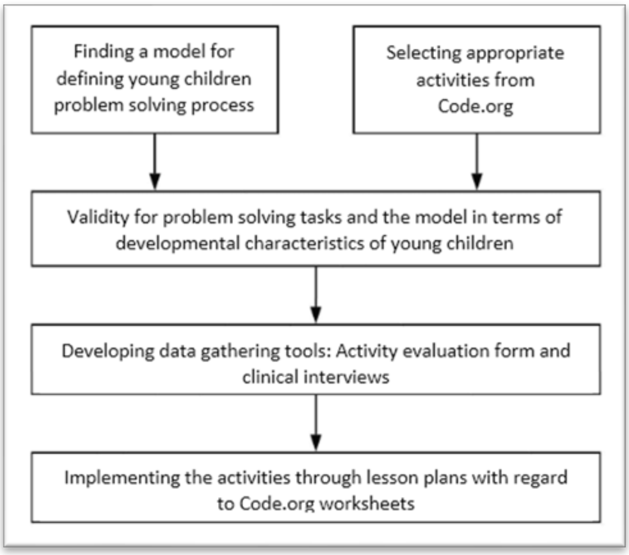
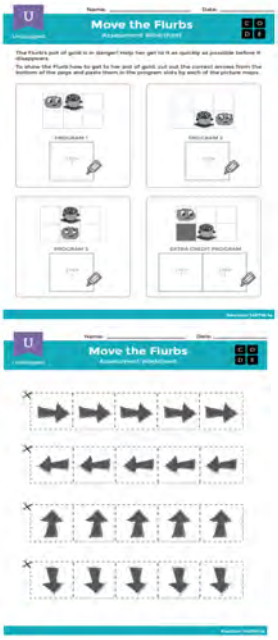


Figure1. Research Procedure

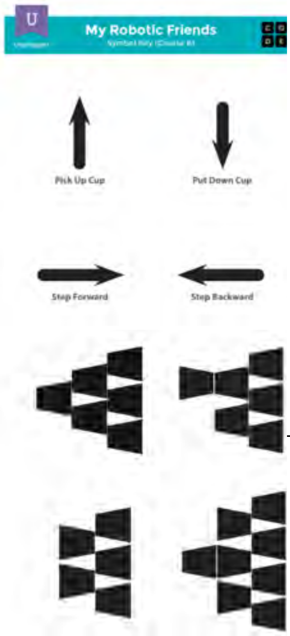
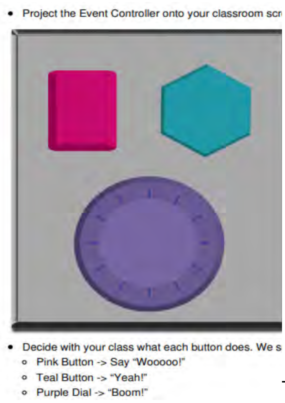
2.3. Selected CS-unplugged Activities for this Research

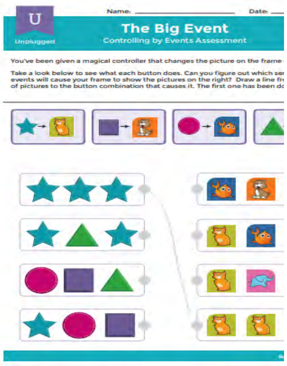
The "36-48 and 48-60-month-old Preschool Students' Developmental Characteristics Guide" was guied us to test the compatibility of the activities with the developmental characteristics of the children (Ministry of Education (MoNE), 2013). Table1 presents students' developmental characteristics and the activities. The tasks in the activities are assigned into four steps (understanding, planning, implementation, and evaluation).

Table1. Activities Associated with Students' Characteristics

Code.org Unplugged Activities	Pre-School Level	Language Development	Cognitive Development	Social Development	Motor Development
1- Happy Maps  	**36-48 Months Old Groups**	Expresses their feelings verbally.	Creates one-to-one matching.	Participates in group games.	Cuts the given simple shapes.
				Follows the rules under adult supervision.	Performs the printing paste operation.
	48-60 Months Old Groups	Answers questions such as Why? How? Who?	Answers questions about the object/person/picture that he/she has seen a short while ago.	Adapts to adult/peer leadership.	Bounces the ball on the ground three times.
		Answers questions about shortly simple stories.	Completes the missing parts in the pictures by looking at the example.		
The activity involves constructing an algorithm that takes the character to the desired goal by cutting out simple shapes about instructions. *CT Skill:* Logical thinking, Algorithmic thinking, Problem solving			Answers questions involving cause-effect relationship.		
2-Real Life Algorithms	**36-48 Months Old Groups**	Expresses their feelings verbally.	Creates one-to-one matching.	Participates in group games.	Cuts the given simple shapes.
		Describes two events in the order in which they occurred. Answers questions	Identifies the object whose picture she sees.	Follows the rules under adult supervision.	Performs the printing paste operation.
			Continues the pattern consisting of two objects by		

 In this activity, the algorithm flow of daily life examples is given in a mixed order and must be ordered correctly. *CT Skill:* Algorithmic thinking, Efficiency, Problem solving		about him/her daily routine.	looking at the model.		
	48-60 Months Old Groups	Answers questions such as Why? How? Who?	Sorts an event in the order in which it occurred.	Adapts to adult/peer leadership.	
		Answers questions about shortly simple stories.	Answers questions about the object/person/picture that he/she has seen a short while ago.		
			Creates a story from the shown pictures.		
			Answers questions involving cause-effect relationship.		
3-Getting Loopy  This activity includes recognizing repetitive steps and performing loops in order of flow with body movements including the language, self-care, cognition and motor development skills of children. *CT Skill:* Algorithmic thinking, Innovative thinking	**36-48 Months Old Groups**	Expresses their feelings verbally.	Shows the parts of their body which are said to her/him.	Participates in group games.	Cuts the given simple shapes.
		Describes two events in the order in which they occurred.		Follows the rules under adult supervision.	Performs the printing paste operation.
	48-60 Months Old Groups	Performs two or three consecutive instructions.	Groups 1-5 objects according to their common properties.	Adapts to adult/peer leadership. Efforts to go on the work he/she started.	Makes simple dance steps.
			Answers questions about the object/person/picture that he/she has seen a short while ago.		
			Completes the missing parts in the pictures by looking at the example.		
			Answers questions involving cause-effect relationship.		

4-My Robotic Friends  This activity includes making the different shapes of towers with plastic cups by following the given instructions. *CT Skill:* Algorithmic thinking, Innovative thinking	**36-48 Months Old Groups**	Expresses their feelings verbally.	Sorts an event in the order in which it occurred.	Participates in group games.	Cuts the given simple shapes.
		Describes two events in the order in which they occurred.		Follows the rules under adult supervision.	Performs the printing paste operation.
				Fulfills simple responsibilities	Builds a tower by 8 cubes.
				Answers questions about him/her.	
	48-60 Months Old Groups	Performs two or three consecutive instructions.	Groups 1-5 objects according to their common properties.	Adapts to adult/peer leadership.	Builds a tower by 10 cubes.
		Performs tasks related to objects out of own sight.	Answers questions about the object/person/picture that he/she has seen a short while ago.		
			Completes the missing parts in the pictures by looking at the example.		
			Answers questions involving cause-effect relationship.		
5-The Big Event 	**36-48 Months Old Groups**	Expresses their feelings verbally.	Creates one-to-one matching.	Participates in group games.	Draws the model shown by looking at the example.
		Performs two or three consecutive instructions.	Identifies the object whose picture she sees.	Follows the rules under adult supervision.	
			Continues the pattern consisting of two objects by looking at the model.		
	48-60 Months Old Groups	It tells what the source of the sound is.	Sorts an event in the order in which it occurred.	Adapts to adult/peer leadership.	
		Answers questions	Groups 1-5 objects according		

 In this activity, the teacher asks the students when the teacher touches different shapes to make different sounds to get the meaning of the shapes. Then, with the same logic, the students are expected to match the shapes' representative animal characters. *CT Skills:* Algorithmic thinking, Innovative thinking, Problem solving, Critical thinking	such as Why? How? Who?	to their common properties.	
	Performs two or three consecutive instructions.	Completes the missing parts in the pictures by looking at the example.	
		Answers questions involving cause-effect relationship.	
		Compares objects according to their various properties.	

2.4. Data Collection Tools

In this study, activity evaluation form and interviews were two main data collection tools.

2.4.1. Activity Evaluation Form (AEF)

AEF was developed for evaluating the problem-solving skills by monitoring the children's behaviors in the problem-solving tasks and considering their perspectives about their experiences. Nance (2016) problem solving steps understanding, planning, doing and looking back were taken as a framework for each activity.

Understand: The researcher asked, "What does this activity ask you to do?" and achievement of the student's ability to understand and verbally express the problem was evaluated.

Plan: The children were asked the question of "What will you do for this activity, what do you need to achieve the result?" to determine their planning of the problem.

Do: Children were asked to perform specific tasks for each activity. For example, in Happy map activity, they were expected to complete the task of the "Finding correct and short way arrows".

Look Back: The questions "Do you think you got this activity right? Do you think what you did was right? Do you have any idea how to fix it if you think you did wrong?" were asked to the children to reveal how they check their solutions.

Considering Nance (2016) framework, each activity was divided into sub-tasks and indicators were defined for each of the tasks. The problem-solving performances were evaluated through these indicators. Students' answers were scored as "satisfactory", "partially-satisfactory" and "unsatisfactory" for each activity. The behaviors of the children in the activity were observed and confirmed with the interview data. Two researchers scored the students' answers in the AEF individually. Then, they discussed the scores together until they came to an agreement about the scores. The scores were also confirmed via the teachers' opinions. So, a triangulation is done with the quantitative and qualitative data handled to reveal the problem-solving skill development of the students.

For instance, in “Understanding” step, AEF was filled for the Activity-1 as described below.

Satisfactory: Using a correct sense of the expected expressions of the activity. For example, *using the arrows to bring the character to the apple* etc.

Partially Satisfactory: Although not emphasizing the expected concepts, short but meaningful expressions are explained. For example, *do not eat apple, do not go to the apple* etc.

Unsatisfactory: Wrong representation of different expressions or failure to fully understanding about the task. For example, *independent expressions or took apples to the character (vice versa).*

A view from the AEF including the “Understanding” step for the Happy map is presented in Table2.

Table2. A view from the activity evaluation form

Activity	**Appropriate Tasks for the event (Based on Nance's problem-solving steps)**	**Evaluation Criteria**		
	Problem Solving Step: Understanding	**Satisfactory**	**Partially-Satisfactory**	**Unsatisfactory**
Happy map activity	*Question (Teacher):* What will you do in this event according to your opinion? What will you do in this game? What does this activity ask you to do? Task-1: To express the logical flow of the game verbally. (saying something similar to “character should go towards apple")	use to "go to toward to apples via using arrows" or use a similar expression	use to "go to toward to apples or eat apples" use a similar but right expression	use to different expressions such as "apple should go to character"

In order to calculate the total scores obtained from all activities, if the “satisfactory” and “partially-satisfactory” scores for the activity were more than “unsatisfactory” scores, the children’s performance was assigned as “satisfactory” for that activity. If the scores obtained from all activities were “partially-satisfactory” and “unsatisfactory”, the score is assigned as “partially-satisfactory” for the activity. If the scores assigned for almost all activities were “unsatisfactory”, the student’s performance was defined “unsatisfactory” for the activity. The criteria for determining the total scores are presented in Table3.

Table 3. Criteria for determining the total programming performance of the children

Evaluation Criteria	**Assigned Scores**
Satisfactory + Partially Satisfactory score is more	Satisfactory
Partially- Satisfactory + Unsatisfactory scored is more	Partially- Satisfactory
Unsatisfactory scored is more	Unsatisfactory

Values from evaluation criteria when analyzing data obtained from this form are scored as Satisfactory=2, Partially Satisfactory =1, Unsatisfactory=0.

2. 4. 2. Interviews

Interviews were conducted one by one and lasted 10-15 minutes. The details of students' artifacts such as "Why did you do that? Do you think you did right? How did you decide?" were asked to children to understand what the student thought when they were doing the tasks. Qualitative data were analyzed via content analysis by transcribing the interviews. To develop categories and codes, two coders read the children responses carefully. The codes were put into categories depending on the programming steps to address the programming performances.

3. Results

In this section, children's problem-solving skills were discussed regarding their performances and their behaviors.

3.1. Problem solving performances in CS unplugged activities

The children's total problem-solving performances by taking into account the evaluation criteria (unsatisfactory, partially satisfactory, and satisfactory) in the AEF were shown in Figure 2.

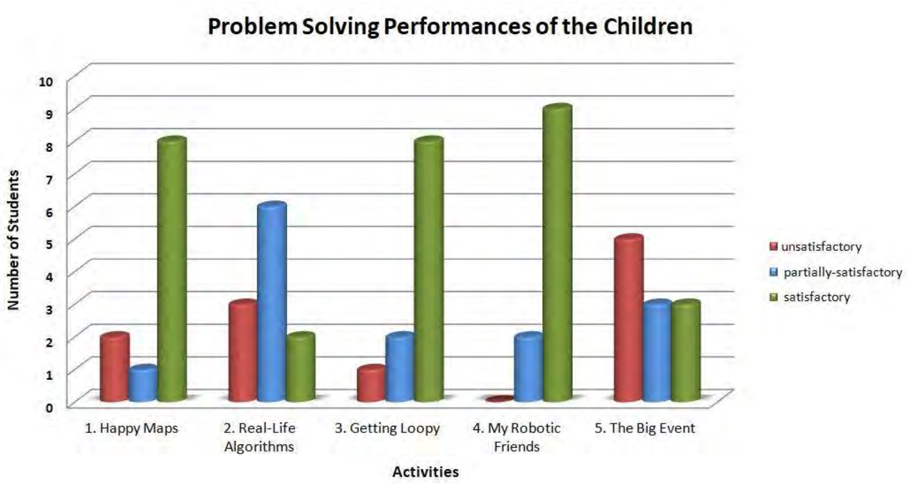


Figure 2. Problem solving performances of the children

Figure2 shows that the lowest average problem solving performance that was observed in Activity 2, and the highest score as 2. The average scores regarding the steps of programming performances are normalized and presented as percentages is shown in Figure 3.

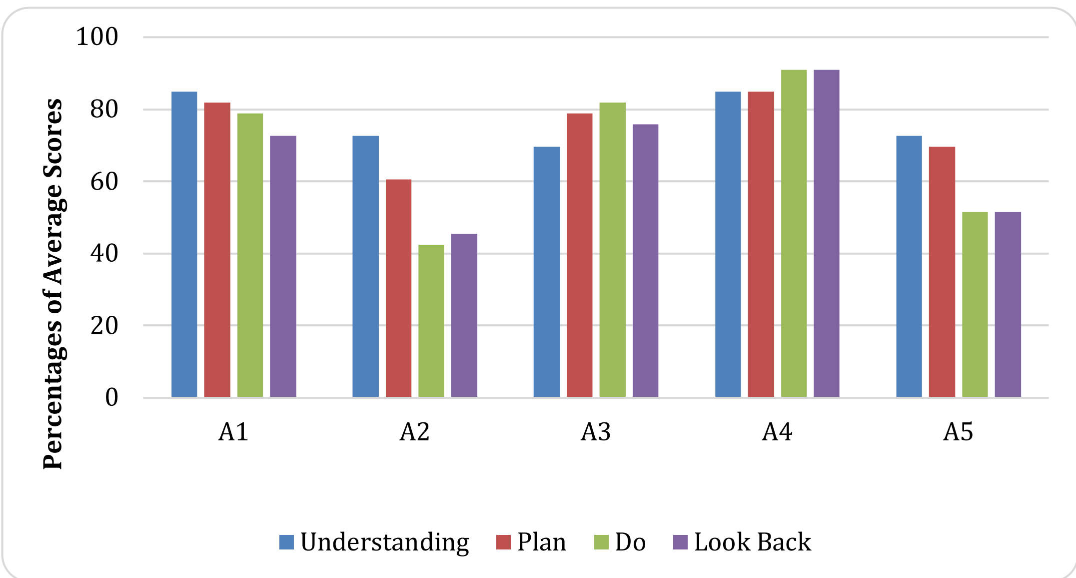


Figure 3. Average achievement scores in the problem-solving steps

Figure 3 shows the percentages of students' average problem-solving performances in all the four sub-steps scores (understanding, plan, do, and look back). For example, for the A1 activity, average score from 11 students was calculated as 2.54, and then this score converted to the percentage of 84.8% for representing A1 activity understanding sub-level. It is seen that, while the average problem-solving performances are high at A1 and A4, and those are relatively low in A2 and A5 which includes mostly ordering and matching activities.

The total problem-solving scores for all activities is shown in Figure 4.

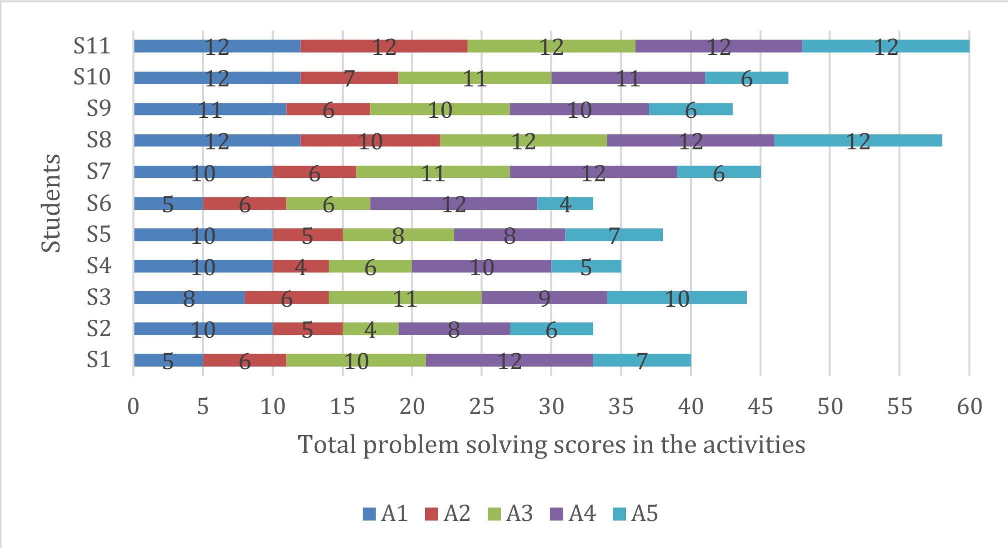


Figure 4. Total problem solving scores in the activities

Figure 4 indicates that the scores from the activities vary for each child. For instance, while S2 got 10 from A1, she got low scores from other activities.

3.2. Students' Experiences in Problem Solving Tasks

In this section, the problem solving performances in the activities are explained with regard to their experiences.

3.2.1. Activity 1. Happy Maps: The performances in the implementation and evaluation steps were slightly lower. In this activity, while determining the correct move within the relevant column for each step, some children *focused on only one column* and *chose all their moves from that column.*

In *planning step:* The majority of the children were able to express their actions in a specific order. However, in *doing step*, instead of completing the tasks as they planned, *they moved away from their planned solution* ways; namely, they did not follow their plans. Some of them attempted to continue with the cross moves. In this sense, S4 expressed that "*I used the short way because I thought that the character was hungry, and it was tired and needed to eat*". This may be related to the imaginations about using an object for different purposes. This kind of imagination is frequently seen in young that they sometimes focus on other objects rather than the goal (Yeşilyaprak, 2018). Some of them also within their imaginations assigned some new meanings for the tasks and acted in the activities like playing games. In this sense, S7 expressed that *"I need to stick the arrows in the right direction and take this character to dinner by following the path (showing with his finger)".* Although he expressed his plan by identifying correct steps, during the activity he tried many wrong ways. He explained this case with this statement "*...I know the right way for the solution, but I wanted him (character) to get confused so I didn't show it*". Similarly, S3 explained the reasons why he did not apply his plan in the activity as *"I didn't want the beast to eat that food, because it always eats all food and is going to be very fat*". In *looking back step*, S6 stated that "*I wanted it to go this way*" and he did not follow the directives and he did not look back to the situation. In general, although the children worked in the tasks as expected in this activity, the labyrinths, which were gradually getting difficult, made it difficult to apply the plans.

3.2.2. Activity2. Real Life: In this activity, children were asked to arrange activities such as tying their shoes, brushing teeth and planting seed in a sequential manner. The tasks include ordering the pictures presented in a mixed order to form an integrity in accordance with the related games. In the Activity 2, the children performed high in *understanding* and *planning* steps, but the average scores taken from the tasks given in the *doing* and *looking back* steps were low. In the tooth brushing activity the children experienced the tasks in their daily lives. Similarly, in the shoe-tying activity as they previously experienced, children got high scores in understanding and planning tasks. Regarding this activity, some of the children when answered the questions like "Why did you put a picture of clean-toothed cat at the end when sorting pictures in this event?", "Do you think you did this activity right?" An example answer is "*... because our teeth are clean when we brush our teeth, so I put it in the end (showing picture of a clean tooth cat).*"

In the seed planting activity, the performance of the children was relatively low especially during the *doing* step. In this activity, it was seen that some of the children copied other children's behaviors while performing the tasks. In addition, the symbols of the seeds used in planting seed and potted seedlings were not clear. It is thought that it is difficult to establish the relations among the pieces of a whole.

It is observed that children's experiences in the tasks significantly influenced their performance particularly in doing step. In fact, children who did not know the symbols in planting seed activity found it difficult to solve the problem. Considering the average scores of the three activities, the lower scores of the children can be thought as a reflection of the lower scores of the doing step. In addition, the fact that the planting seed (6 stages) activity was completed in more stages than shoe-tying (3 stages), tooth brushing (4 stages) activities may have influenced the low performance in implementing their plans.

3.2.3. Activity3. Getting Loopy: In this Activity, performances related to the tasks given for the *understanding* and *doing* steps were high but low at the *looking back* step. It was observed that all of the children in their plans provided repeated some actions (loop). However, the majority of the children could not determine the number of repetitions in this process, and they could not show the number of loops in the plans correctly. In this sense, S5 addressed that *"...I will repeat clapping, clapping, clapping as my teacher doing*" while repeating processes. Another student S3 stated, "*…I'll repeat the same picture, but I don't know how many times*".

On the other hand, the tasks in the doing step were done by following the teacher's presentation. The presentations helped children to get high scores even though the children had deficiencies in their plans. At the looking back step, some of the children could not perform the tasks by assigning the number of repetitions correctly.

3.2.4. Activity4. My Robotic Friends: In this activity, the children were asked to put the cups in order as in the pattern given on the worksheet. A view from students' actions is seen in Figure 5.

Figure 5. A view from My Robotic Activity

In this activity, almost all of the children showed high performance in *understanding* and *planning*, and all of the children were successful in *doing* and *looking back*. For instance, S4 expressed that "*… I will put the cups in order as shown in picture", "We were like robots in this activity*". Also, a number of children were able to decide the number of cups for the correct solution in the planning step. In addition, it is seen that in all of the tasks in the doing step, the children were able to put the cups in order as expected pattern. Using concrete object such as plastic cups may be considered as one of the reasons of high performance in this step. Using daily-recognized objects and allowing these objects to create the patterns by heuristic approach might have been contributed to this achievement. Accordingly, S1 identified that "sometimes *I'm confused while I'm putting the cups in order, but then I just lined up like a tower."* One other reason for the high performance of the children at the looking back step may be related to the developing a concrete product. In this sense, the children could review the paths they needed to follow when they could not do it correctly.

3.2.5. Activity5. Big Event: In this activity, the children were asked to match the geometric shapes with the appropriate animal figures.

Figure 3 shows that the majority of the children performed high in *understanding* and *planning* steps in Activity5. The performances in the doing and looking back steps were quite low. The average scores in the *doing* and *looking back* steps were lower than the other steps. Children's perspectives reflect that one reason for the low scores may be the fact that the children cannot remember more than one pattern. For instance, S5 expressed that "*...When I forgot the pattern, I looked up again and I waited to put my finger on it to not forget"* S4 also explained his action as "*...I've confused which shape corresponds to which animal"*. As in other activities, it can be thought that the decrease in the performance of the problem solving is regarded to the complexity of cognitive tasks such as keeping the one more pattern in mind and creating multiple patterns. In addition, in this activity it is expected that while matching, first; understand the hint box given in the introduction, and then keep this information in mind, then adapt the following question according to the situation, and lastly, use this information to reach the desired result. Since this activity requires sequential follow-up and mental processes, it is not easy to perform the expected tasks in this age group sequentially.

Figure 3 indicates that in Activity 1, Activity 2 and Activity 5, the understanding steps were completed more successfully than the other steps, whereas the doing step were more successful than the other steps in the Activity 3 and Activity 4. It is seen that the planning step is constructed accordingly depending on how well the student completes the understanding step. It is noteworthy that children performed lower in the looking back steps except for Activity 2 and Activity 4 than other steps. At this point, the result may be about the nature of the activities Activity 2 and Activity 4. Because Activity 2 is more directly related to the daily life than others and Activity 4 (My robotic friends) addressed more motor skills.

4. Discussion

This study attempted to determine the effect of CS-unplugged activities on problem solving skills of preschool children. The results indicated that the significant increase in problem solving skills may be due to the activities designed in accordance with the learning objectives. In this study, in all of the activities the children performed high in understanding and planning steps compared to the other steps. The doing step, which is usually used to cut, paste, match, sort, etc. resulted in lower scores due to cognitive skills as well as hand skills. It is surprising that although the scores during the looking back steps were low, the children began to perform the tasks correctly in this step.

The current study confirmed that CS-unplugged activities can support the young children to establish a relationship between activity and real life. Namely, the activities including concrete events such as putting the cups in order together activity can provide high performance rates at the looking back steps. In Activity 2, although the children did not experience a problem before, they could understand the task, but mostly they could not perform high scores in doing step. In this context, one can infer that the activities that the children experienced before can support children' performances in doing step as in the understanding step. For instance, in Activity 1 (Happy Maps) and Activity 5 (Big Event), children's understanding and planning performances were high, but they could not perform the similar performance in the doing and looking back steps. In these two activities, keeping multiple moves in mind, matching multiple images and performing sequential operations are some examples requiring the advanced cognitive skills in which the children could understand the problem but not perform high in the doing step.

The findings showed that study concluded that the design of the activities and the roles attained to the children influenced the development of their problem-solving skills. Similarly, another study suggested that the activities in CS-unplugged activities should be explicitly linked to central concepts in CS (Taub, Armoni, & Ben-Ari, 2012). In accord to this study, Faber et al. (2017) found that the unplugged materials seem to elicit positive reactions from children. Another reason for the achievement in activities is the nature of the activities that preschool students are generally considered to have high performance due to their similarity to cut-and-paste activities. In accord to this finding; a comparative experiment by Montes-Leon et al. (2020) found that the introduction of unplugged programming activities could help students improve their CT skills and have a positive effect on their follow-up programming learning.

It is important for children to engage in the tasks of problem-solving activities. Actually, it is also known that the attention of the young children can be distracted quickly, and it is difficult to engage them in different tasks during the activities (Radesky & Christakis, 2016; Rodriguez et al, 2016). In the current study, attractive potentials of the tasks in CS-unplugged can be seen as engaging children in the activities and being active in problem solving. As suggested in prior studies, we ensured that all children had a role to act in the activities (any amount of down time potentially results in bored, disengaged children and lower assessment results).

Research has shown that unplugged programming activities can effectively develop 5–7-year-old children CT skills and help them transfer CT skills to other problem-solving scenarios (Conde et al., 2017; Wohl et al., 2015). Unplugged debugging activities without the use of programming tools provide more content-focused learning experiences for younger students by reducing the cognitive demands for using technological tools (Kotsopoulos et al., 2017). This study also confirmed that CS-unplugged activities including objects or concepts that the children have experienced before positively influence on their planning and doing performances. The results also indicated that, the activities should also be designed considering the "imaginary world" of the children. Namely, something in the activity may remind them of some different events in their mind. Hence, the activities including tangible and basic materials may eliminate this and may provide better problem-solving outcomes.

This study highlights that CS-unplugged activities may provide successful outcomes for problem solving of preschool students. The preschool children can engage cognitively, socially, and creatively in the CS-unplugged activities. In this study we separated the problems into the tasks by associating them with the substages of problem solving. In this way, we could evaluate the problem-solving processes regarding their achievements in the tasks in the understanding, planning, doing and looking back steps. With both findings provide potential avenues for future problem solving, this study moves us one step closer to uncovering a way to evaluate the young children's problem-solving process.

This research is not exempt from limitations. The most important of which is its exploratory nature. It is difficult to provide quantitative data about the young children's problem-solving processes. It should be noted that this study focused on only 5 activities to evaluate the problem-solving performances in CS-unplugged activities. Implementing activities by taking objectives in the preschool curriculum and students' developmental characteristics into consideration played a positive role about the implementation process. This study used only students' answers and observations assess the problem-solving processes. In future studies, data from the video records including the interactions among children would support evaluating the problem-solving processes more accurately. Moreover, Taub, Armoni and Ben-Ari (2012) pointed out that it is difficult to demonstrate that CS-unplugged activities actually achieve long-term goals about directing young children's interest in CS concepts. Hence, further longitudinal studies may be helpful in clarifying the effect of CS-unplugged activities to the achievements in CS. One other limitation was the small sample size; thus, a larger sample size would increase the sensitivity of the analysis.

5. Conclusion

This study considered the cognitive development of the young children by directing the roles of children to problem solving and evaluated developments in their problem-solving skills. The results indicate that even if the children's plans about the tasks is correct, sometimes the problem-solving process cannot be fully completed as expected in the doing and looking back steps. The tasks in the activities were also found influential on achieving problem solving steps. Preschool children's developmental and working memory characteristics and their previous experiences about the objects and the tasks in the activities also influenced their problem-solving process.

Overall, the contribution of the findings of this study is in two folds. One is about the design attributes of the problem-solving activities for preschool children. The second is about the evaluation process of CS-unplugged activities in terms of problem solving. It is recommended to design worksheets that are both engaging for the children and directing them to problem solving process. Incorporating worksheets or assessment techniques into lesson plans of preschools is also crucial to take the advantage of problem solving in the early ages. Instructional designers should take care when deciding to design certain types of learning activities considering children's developmental characteristics. Educators can adapt CS-unplugged activities to their lessons are to build and maintain a collaborative classroom environment and refer them when teaching abstract concepts and solving daily problems in preschool classrooms. We hope that the findings of this study would assist in future design and implementation of CS- unplugged activities for young children.

References

Ahn, J., Sung, W., & Black, J. B. (2021). Unplugged debugging activities for developing young learners' debugging skills. *Journal of Research in Childhood Education*, 1-17. https://doi.org/10.1080/02568543.2021.1981503

AlAmer, R. A., Al-Doweesh, W. A., Al-Khalifa, H. S., & Al-Razgan, M. S. (2015). Programming unplugged: bridging CS unplugged activities gap for learning key programming concepts. In *2015 Fifth International Conference on e-Learning (econf)* (pp. 97-103). IEEE. https://doi.org/10.1109/ECONF.2015.27

Barr, D., Harrison, J. and Conery, L. (2011). Computational thinking: A dijital age skill for everyone. *Learning & Leading with Technology*, *38*(6), 20-23. Google Scholar

Battal, A., Afacan Adanır, G., & Gülbahar, Y. (2021). Computer Science Unplugged: A Systematic Literature Review. Journal of Educational Technology Systems, *50*(1), 25-47. https://doi.org/10.1177/00472395211018801

Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2009). Computer science unplugged: School students doing real computing without computers. *The New Zealand Journal of Applied Computing and Information Technology*, *13*(1), 20-29. Google Scholar

Bell, T., & Vahrenhold, J. (2018). CS unplugged-how is it used, and does it work? In H.J.Böckenhauer, D. Komm & W. Unger (Eds.), *Adventures between lower bounds and higher altitudes* (pp. 497–521). Springer. https://doi.org/10.1007/978-3-319-98355-4_29

Bell, T., Witten, I., & Fellows, M. (2015). *CS Unplugged: An enrichment and extension programme for primary-aged students.* Retrieved 10 April 2018 from http://csunplugged.org/wp-content/uploads/2015/03/CSUnplugged_OS_2015_v3.1.pdf

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

Brown, Q., Mongan, W., Kusic, D., Garbarine, E., Fromm, E. and Fontecchio, A. (2013). *Computer aided instruction as a vehicle for problem solving: Scratch programming environment in the middle years classroom*. *Proceedings of the Annual Conference and Exposition,* Pittsburg, Pensilvania USA. Google Scholar

Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A., & Barone, D. (2017, November). Development of computational thinking skills through unplugged activities in primary school. In *Proceedings of the 12th Workshop on Primary and Secondary Computing Education* (pp. 65-72). ACM. https://doi.org/10.1145/3137065.3137069

Bransford, J. D., Zech, L., Schwartz, D., Barron, B., Vye, N., & CTGV. (2000). Designs for environments that invite and sustain mathematical thinking. In Cobb, P. (Ed.), *Symbolizing, communicating, and mathematizing: Perspectives on discourse, tools, and instructional design.* (pp. 275-324). Mahwah, NJ: Erlbaum

Caelien, E. N., & Yadav, A. (2020). Unplugged approaches to computational thinking: A historical perspective. *TechTrends, 64*(1), 29-36. https://doi.org/10.1007/s11528-019-00410-5

Conde, M. A., Ferna ´ndez-Llamas, C., Rodrıguez-Sedano, F., Guerrero-Higueras, A. M., Matella´n-Olivera, V., & Garcıa-Penalvo, F. J. (2017). Promoting computational thinking in K - 12 students by applying unplugged methods and robotics. In J. M. Dodero, M. S. I. Sa´iz, & I. R. Rube (Eds.), *Proceedings of the Fifth International Conference on Technological Ecosystems for Enhancing Multiculturality* (TEEM 2017). Ca´diz, Spain: ACM. https://doi.org/10.1145/3144826.3145355

Creswell, J. W., Plano Clark, V. L., Gutmann, M. L. and Hanson, W. E. (2003). Advanced mixed methods research designs. In A. Tashakkori, C. Teddlie (Eds.), *Handbook of mixed methods in social and behavioral research* (pp.209–240). Thousand Oaks, CA: Sage,

Curzon, P. (2014). Unplugged computational thinking for fun. Retrieved April 25, 2020 from: https://publishup.uni-potsdam.de/opus4-ubp/frontdoor/deliver/index/docId/8257/file/cid07_S15-27.pdf

Çetin, E. (2016). *A case study for the use of technology aided graphical organizers in preschool children's problem solving process*. Unpublished PhD. Dissertation, Gazi University, Ankara.

Daan, W., Cooper, S., & Pausch, R. (2009). *Learning to Program with ALICE*. Pearson education. Google Scholar

Dagien˙e, V., Stupurien˙e, G., Vinikien˙e, L. (2016). Promoting inclusive informatics education through the Bebras Challenge to all K-12 students. In: *Proceedings of the 17th International Conference on Computer Systems and Technologies 2016* (pp. 407–414). ACM. https://doi.org/10.1145/2983468.2983517

Del Olmo-Muñoz, J., Cózar-Gutiérrez, R., & González-Calero, J. A. (2020). Computational thinking through unplugged activities in early years of Primary Education. *Computers & Education, 150*, 103832. https://doi.org/10.1016/j.compedu.2020.103832

Dwyer, H., Hill, C., Carpenter, S., Harlow, D., & Franklin, D. (2014). Identifying elementary students' pre-instructional ability to develop algorithms and step-by-step instructions. In *Proceedings of the 45th ACM Technical Symposium On Computer Science Education* (pp. 511-516). ACM. https://doi.org/10.1145/2538862.2538905

Faber, H. H., Wierdsma, M. D., Doornbos, R. P., van der Ven, J. S., & de Vette, K. (2017). Teaching computational thinking to primary school students via unplugged programming lessons. *Journal of the European Teacher Education Network*, *12*, 13-24.

Gardeli, A., & Vosinakis, S. (2017). Creating the computer player: An engaging and collaborative approach to introduce computational thinking by combining unplugged activities with visual programming. *Italian Journal of Educational Technology, 25*(2), 36–50. https://doi.org/10.17471/2499-4324/910

Grgurina, N., Barendsen, E., Zwaneveld, B., van Veen, K., & Stoker, I. (2014). Computational thinking skills in Dutch secondary education: exploring pedagogical content knowledge. In *Proceedings of the 14th Koli Calling International Conference on Computing Education Research* (pp. 173-174). ACM. https://doi.org/10.1145/2674683.2674704

Grout, V., & Houlden, N. (2014). Taking computer science and programming into schools: The Glyndŵr/BCS Turing project. *Procedia-Social and Behavioral Sciences*, *141*, 680-685. https://doi.org/10.1016/j.sbspro.2014.05.119

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

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

Hermans, F., Aivaloglou, E. (2017): To scratch or not to scratch?: a controlled experiment comparing plugged first and unplugged first programming lessons. In *Proceedings of the 12th Workshop on Primary and Secondary Computing Education*, WiPSCE 2017, pp. 49–56. ACM, New York. https://doi.org/10.1145/3137065.3137072

Hodhod, R., Khan, S., Kurt-Peker, Y., & Ray, L. (2016). Training teachers to integrate computational thinking into K-12 teaching. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education* (pp. 156-157). ACM. https://doi.org/10.1145/2839509.2844675

Huang, W., & Looi, C. K. (2021). A critical review of literature on "unplugged" pedagogies in K-12 computer science and computational thinking education. *Computer Science Education, 31*(1), 83-111. https://doi.org/10.1080/08993408.2020.1789411

Jagušt, T., Krzic, A. S., Gledec, G., Grgić, M., & Bojic, I. (2018). Exploring different unplugged game-like activities for teaching computational thinking. In *2018 IEEE Frontiers in Education Conference (FIE)* (pp.1-5). IEEE. https://doi.org/10.1109/FIE.2018.8659077

Kafai, Y. B., & Burke, Q. (2013). Computer programming goes back to school. *Phi Delta Kappan*, *95*(1), 61-65. https://doi.org/10.1177/003172171309500111

Kalelioğlu, F., & Keskinkılıç, F. (2017). Teaching methods for computer science education. In Y. Gülbahar (Ed.), *From computational thinking to programming* (1st ed. pp. 155–182). Ankara.

Kalelioğlu, F., Gülbahar, Y., Akçay, S., & Doğan, D. (2014). Curriculum integration ideas for improving the computational thinking skills of learners through programming via scratch. In *Local Proceedings of the 7th International Conference on Informatics in Schools: Situation, Evolution and Perspectives* (pp. 101-112). Google Scholar

Kotsopoulos, D., Floyd, L., Khan, S., Namukasa, I. K., Somanath, S., Weber, J., & Yiu, C. (2017). A pedagogical framework for computational thinking. *Digital Experiences in Mathematics Education, 3*(2), 154-171. https://doi.org/10.1007/s40751-017-0031-2

Lambert, L., & Guiffre, H. (2009). Computer science outreach in an elementary school. *Journal of Computing Sciences in Colleges*, *24*(3), 118-124. Google Scholar

Lee, I., Martin, F., & Apone, K. (2014). Integrating computational thinking across the K—8 curriculum. *Acm Inroads*, *5*(4), 64-71. https://doi.org/10.1145/2684721.2684736

Leifheit, L., Jabs, J., Ninaus, M., Moeller, K., & Ostermann, K. (2018). Programming unplugged: An evaluation of game-based methods for teaching computational thinking in primary school. In M. Ciussi (Ed.), *ECGBL 2018 12th European Conference on Game-Based Learning* (p. 344). Academic Conferences and Publishing Limited. Google Scholar

[Milli Eğitim Bakanlığı (MEB)], 2013. Okulöncesi eğitim programı. Retrieved 10 May 2018 from http://tegm.meb.gov.tr/dosya/okuloncesi/ooproram.pdf

Montes-León, H., Hijón-Neira, R., Pérez-Marín, D., & Montes-León, R. (2020). Mejora del pensamiento computacional en estudiantes de secundaria con tareas unplugged. *Education in the Knowledge Society*, 21, 24. Google Scholar

Nance, S. (2016). *Using computer programming to enhance problem-solving skills of fifth grade students*. Unpublished PhD. Thesis, University of Florida, Gainesville.

Nishida, T., Idosaka, Y., Hofuku, Y., Kanemune, S., & Kuno, Y. (2008). New methodology of information education with "Computer science unplugged". In *International Conference on Informatics in Secondary Schools-Evolution and Perspectives* (pp. 241-252). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-69924-8_22

Piaget, J. (1954). *The construction of reality in the child* (M. Cook, Trans.). New York, NY, US.

Piaget, J. (1976). Piaget's theory. In *Piaget and his school* (pp. 11-23). Springer, Berlin, Heidelberg.

Prottsman, K. (2014). Computer science for the elementary classroom. *ACM Inroads*, *5*(4), 60-63. https://doi.org/10.1145/2684721.2684735

Radesky, J. S., & Christakis, D. A. (2016). Keeping children's attention: The problem with bells and whistles. *JAMA pediatrics, 170*(2), 112-113. https://doi.org/10.1001/jamapediatrics.2015.3877

Repenning, A., Webb, D., & Ioannidou, A. (2010). Scalable game design and the development of a checklist for getting computational thinking into public schools. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education* (pp. 265-269). ACM. https://doi.org/10.1145/1734263.1734357

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... and Kafai, Y. (2009). Scratch: programming for all. *Communications of the ACM*, *52*(11), 60-67. Google Scholar

Rodriguez, B. R. (2015). *Assessing computational thinking in computer science unplugged activities.* Unpublished Masters thesis. Golden: Colorado School of Mines.

Rodriguez, B., Rader, C., & Camp, T. (2016). Using student performance to assess CS unplugged activities in a classroom environment. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education* (pp. 95-100). ACM. https://doi.org/10.1145/2899415.2899465

Taub, R., Ben-Ari, M., & Armoni, M. (2009). The effect of CS unplugged on middle-school students' views of CS. *ACM SIGCSE Bulletin, 41*(3), 99–103. https://doi.org/10.1145/1595496

Taub, R., Armoni, M., & Ben-Ari, M. (2012). CS unplugged and middle-school students' views, attitudes, and intentions regarding CS. *ACM Transactions on Computing Education (TOCE)*, *12*(2), 8. https://doi.org/10.1145/2160547.2160551

Thies, R., & Vahrenhold, J. (2012). Reflections on outreach programs in CS classes: learning objectives for unplugged activities. In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education* (pp. 487-492). ACM. https://doi.org/10.1145/2157136.2157281

Uchida, Y., Matsuno, S., Ito, T., & Sakamoto, M. (2015). A proposal for teaching programming through the five-step method. Journal of Robotics, *Networking and Artificial Life,* *2*(3), 153. https://doi.org/10.2991/jrnal.2015.2.3.4

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, *20*(4), 715-728. https://doi.org/10.1007/s10639-015-9412-6

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33-35. Google Scholar.

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

Wing, J. (2011). Research notebook: Computational thinking-What and why? *The Link Newsletter, 6, 1–32. Retrieved from http://link.cs.cmu.edu/files/11-399_The_Link_Newsletter-3.pdf.* Google Scholar.

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

Yeşilyaprak, B. (2006). *Education Psychology*. Pagema Publisher, 376s, Ankara.

Zur Bargury, I., Haberman, B., Cohen, A., Muller, O., Zohar, D., Levy, D., & Hotoveli, R. (2012, October). Implementing a new Computer Science Curriculum for middle school in Israel. In *2012 Frontiers in Education Conference Proceedings* (pp. 1-6). IEEE. https://doi.org/10.1109/FIE.2012.6462365

Appendix1

Completes puzzle with 10-25 pieces.

Creates new shapes by combining geometric shapes

Groups the objects

Performs addition and subtraction operations of at least 1-10 numbers.

Explains how to do match, associate and group Establish cause-effect relationships

Uses comparison statements