Jurnal Pendidikan Usia Dini
http://journal.unj.ac.id/unj/index.php/jpud
Volume 15. Number 1. April 2021
e-ISSN (Online Media): 2503-0566
P-ISSN (Print Media): 1693-1602

Unplugged Coding Activities for Early Childhood Problem-Solving Skills

Mutoharoh[1]
Universitas Sultan Ageng Tirtayasa, Indonesia[1,3,4]
Ahmad Hufad[2]
Universitas Pendidikan Indonesia[2]
Maman Fathurrohman[3]
Isti Rusdiyani[4]

DOI: https://doi.org/10.21009/JPUD.151.07
Accepted: January 15th2021.Approved: March 4th2021. Published: 30th April 2021

ABSTRACT: Problem solving skills are very important in supporting social development. Children with problem solving skills can build healthy relationships with their friends, understand the emotions of those around them, and see events with other people's perspectives. The purpose of this study was to determine the implementation of playing unplugged coding programs in improving early childhood problem solving skills. This study used a classroom action research design, using the Kemmis and Taggart cycle models. The subjects of this study were children aged 5-6 years in Shafa Marwah Kindergarten. Research can achieve the target results of increasing children's problem-solving abilities after going through two cycles. In the first cycle, the child's initial problem-solving skills was 67.5% and in the second cycle it increased to 80.5%. The initial skills of children's problem-solving increases because children tend to be enthusiastic and excited about the various play activities prepared by the teacher. The stimulation and motivation of the teacher enables children to find solutions to problems faced when carrying out play activities. So, it can be concluded that learning unplugged coding is an activity that can attract children's interest and become a solution to bring up children's initial problem-solving abilities.

Keywords*: Early Childhood, Unplugged Coding, Problem solving skills*

[1] Corresponding Author:
Universitas Sultan Ageng Tirtayasa
Email: 777220001@untirta.ac.id

1 INTRODUCTION

In 2012, the United States National Research Council defined 21st-century skills as the basis for the learning process (Pellegrino, J. W., & Hilton, 2012). These skills are divided into three competency domains, such as Interpersonal, Intrapersonal, and cognitive. Cognitive domains include critical thinking, creativity, executive functioning and problem solving skills (Gretter, S., & Yadav, 2016; Papanastasiou et al., 2018) Problem solving skills have an important role for children in supporting social development in society. Problem solving skills can be defined as conflict prevention in social life, resulting in strategies that can be useful for self-development (Berk, 2013). A person can understand the emotions, thoughts and behaviour of others, put oneself in their place, except what they are, respect themselves and develop empathy as part of their personality (Koksal Akyol, A. ve Didin, 2016).

Prior to coding, problem-solving techniques such as planning and modelling must be implemented, they are considered foundational of computer science curriculum and play a critical role in good coding outcomes (Fee & Holland-Minkley, 2010). Coding necessitates the development of problem-solving solutions, the ability to convert proposals into precise executional commands, and the analytical abilities required to assess the effects of these commands (Ismail et al., 2010). The efficient application of techniques also involves approaches for organizing work in a step-by-step logic, evaluating specific cases such as input and output, and formulating the steps for problem solving (Ismail et al., 2010). To put it another way, coding challenges necessitate the use of both problem-solving techniques and program construction skills (Hazzan et al., 2011).

Coding is a process of making step-by-step directions that are carried out in detail and then interpreted and followed up by machines. Actions or steps to reach the target are called Al-ghoritme (McLennan, 2017). Coding or coding system is not new in society and the lives of children, this coding is encountered every day (cell phones, automatic doors, robot vacuum cleaners, etc.). In practice, coding involves a variety of early math skills, science, including spatial, number of, problem solving, inquiry and reasoning. In addition, coding involves communication skills. Coding and decoding (interpreting and understanding) is processes that occur simultaneously. When children make code, they will also break the code. Likewise, when children talk, they understand what they are talking about. Coding in early childhood has an effect on attitudes of knowledge, skills in various

fields such as problem solving and computational thinking (Sullivan, A., & Bers, 2016). Problem solving strategies, such as planning and design, need to be applied before coding as the basis for the fundamentals of science education. computers, because coding requires the skills to convert plans into accurate execution orders, and analytical skills to evaluate results and orders (Deek, 1999;Ismail et al., 2010)

Unplugged coding activity programs or coding that are not connected in learning, use direct activities with a concrete representation of interrupted activities without computers (Joohi Lee, 2019). This is a new teaching method in the 21st century (Threekunprapa, A., & Yasri, 2020) besides being a new teaching method in various countries. Including the US, UK, New Zealand, Germany, India, Georgia, France, Korea, Japan, Sweden, Finland, Israel, Russia and Italy have included computer science as a core subject of learning (Grover, S., & Pea, 2013)**.** Science computational thinking (CT) important for problem solving, which refers to the thought process in defining problems and proposing solutions (Shute, V. J., Sun, C., & Asbell-clarke, 2017) for early childhood unplugged coding can develop children's computational thinking (Brackmann et al., 2017)**.**

Early childhood is the important period in which children discover their world while playing, they acquire new knowledge through their experiences (Sullivan, A. A., Bers, M. U., & Mihm, 2017). Based on the problem of the lack of problem-solving skills in early childhood, and the low level of implementation of the unplugged coding program in early childhood learning, many related studies have revealed its urgency. Therefore, this classroom action research aims to determine whether unplugged coding learning activities can improve children's problem-solving skills.

2 THEORITICAL STUDY

2.1 *The Urgency of Learning Activities Coding for Computational Thinking*

Coding or programming as the language of the digital age consists of a process of developing instructions that are understood and needed by computers so that the program can work (McLennan, 2017; Vorderman, 2017). Coding is not just a programming language, but coding is a way to develop various skills (Harrop, 2018) When coding children learn to solve problems, mathematicians, engineers, storytellers, innovators, and collaborate. Coding helps children to stimulate initial thinking skills. It also supports analytical

thinking and creativity (Akyol-Altun, 2018;Bers, 2018). Various approaches to solving problems in computational thinking can be applied in everyday life. For example, how to arrange books in order, play sudoku, play chess, plan trips, arrange schedules. Even reading detective and mystery stories will be more fun if we have a logical way of thinking (Yohanes, 2018).

For kids, there are a variety of coding environments to choose from (e.g., Logo language, Toon Talk, Squeak Toys, Stage cast Creator, Microworlds JR and Scratch). These prototypes all have features that are particularly well tailored to the developmental stages of early childhood. Simplified grammar, the use of graphic artifacts rather than text, drag-and-drop style, and so on. Via specially developed apps, even pre-school children will learn basic programming skills (e.g., code.org, scratch, etc.). Children are asked to name the actions that refer to the instructions, identify the events in a sequential order, and even create basic programs to achieve their objectives in these visual coding environments (Brennan & Resnick, 2012). Additional experimental trials on the subject centred on young children's attendance and offer additional data relating coding education to effective learning outcomes (McClure et al., 2017). Pila et al., (2019) conducted research on children aged 4–6 who enrolled in a week-long coding camp.

Command learning with "Daisy the Dinosaur" and basic visual coding with "Kodable" were the focus of the coding environments. The participants gained knowledge of Daisy commands and developed their overall coding abilities, according to the study's findings. Sheehan et al., (2019) conducted a survey of 31 parents and children ranged in age from 4 to 5 years old, using a coding education application. The Scratchers coding application can provide a rich framework for STEM learning, and parent–child engagement has a positive impact on student learning outcomes, according to one of the study's main findings.

This concept has evolved over time to incorporate the role of agents in implementing these solutions. The thought processes involved in formulating a problem and communicating its solution in such a way that a program – human or machine – can efficiently carry out" are known as computational thinking (Aranda & Ferguson, 2018). The use of computing machines such as computers and robotics is implicit in this concept, as is the fact that humans can play the part of the machine and execute programs.

Multiple elements have been specifically applied to the body of Computational Thinking literature since Wing (2006). Angeli et al., (2016) suggest a five-part conceptual structure for Computational Thinking (i.e., abstraction, generalization, decomposition, algorithms, and debugging). Hsu et al., (2018) identified 19 Computational Thought thinking stages through experiments, including pattern recognition, algorithm design, and simulation, to name a few.

Remember that children in the preoperational stage will describe concepts and activities using symbols (e.g., pictures and words) (Sigelman & Rider, 2012). After adequate preparation, they should be able to (1) identify patterns involving symbols and (2) present sequences and algorithm designs using symbols or plain terms. Children in the preoperative period, on the other hand, rely on their experiences to solve problems (Sigelman & Rider, 2012). As a result, tangible materials can be used to help them develop Computational Thinking skills. One approach is to have unplugged (no devices) Computational Thinking tasks first, followed by their wired (with devices) counterparts (Looi et al., 2018).

Pre-schoolers will provide a stronger basis for fostering Computational Thinking in wired environments if they provide further specific examples with Computational Thinking in unplugged practices. Teachers will implement algorithm design using the Computational Thinking skills and required vocabulary learned in the above unplugged exercises. The hope is that this concrete experience can expand or migrate to more general and abstract contexts in due time (Armoni, 2012).

2.2 *Unplugged Coding Activities in Early Childhood Education*

Early childhood programming is becoming increasingly relevant. In this sense, educational curricula around the world, including in Turkey, have undergone significant reforms to provide programming education in elementary school (DemiRer & Sak, 2016). Both plugged and unplugged tasks are included within the programming education framework. Unplugged programming leads to learning abstract thought and computer science techniques without the use of computational machines, whereas plugged programming uses computational devices in these learning processes (Aranda & Ferguson, 2018). Role-playing, manipulating real-world objects (such as sticky notes, cards, and wooden

blocks), and actual movements of the body are all examples of unplugged code (Aranda & Ferguson, 2018).

Unplugged Programming is a term that refers to studying coding and computer science principles without the use of computers. Role-playing, manipulation of real-world objects (e.g., post-it sticky notes, cards, wooden blocks), and physical actions of the of among other methods, can be used to accomplish this. Bell et al., (2009) argues that learning this way is not merely about simulating computing systems, but also about giving students the skills to investigate the basic ideas of computer science without the technological skills needed to code. Real-life materials and experiences will support learning in the pre-operational stage of cognitive development. (Wang, D., Zhang, C., & Wang, 2010)

Alghorithms are an unplugged coding activity that can stimulate initial problem solving abilities (Mittermeir, 2013; Futschek, G., and Moschitz, 2010; Voronina et al., 2016). When playing unplugged coding it requires the ability to organize, analyse data then solve problems by dividing it into small pieces so that they can be managed. Another Unplugged coding activity that children can play is playing origami. Origami is perfect for the perfect unplugged coding activity.

Unplugged coding of flowcharts is one way to make code more conceptually precise while retaining the importance of visualization. Unplugged coding exercises use game-based learning to facilitate critical reasoning, with learners being assigned the task about sequentially thinking of an algorithm without using a machine. The use of game elements in learning has been shown to improve cognitive engagement, ability to learn, and deeper reflection on learning experiences (Zvarych et al., 2019). It is also a social space in which students can engage in constructive learning and collaborate with others, such as peers and facilitators (Yuksel, 2019).

Flowcharts are visual representations of a method, procedure, or machine algorithm. Advanced computer science principles such as loops and conditional loops can be cultivated by using various shapes of diagrams to express different interpretations of programming commands. This will compensate for the absence of visual language programming. According to research, this technique will improve coding precision and help students understand the algorithm faster when they receive visual expressions (Scanlan, 1989). Furthermore, it has been seen to be more sophisticated in promoting Computational

Thinking skillsets than pseudocodes, which focus on human reading rather than computer reading.

2.3 *Early Problem-Solving Skills*

Childhood is one of the critical periods of life which is the basis of life, laying basic knowledge and skills through habituation (Arı, 2003). Important problem solving skills are given since preschool and basic education because they are harvesters (Aydoğan, 2004). The importance of improving problem solving skills so that children are able to overcome potential problems in the future and develop them as needed (Anlıak, Ş., & Dinçer, 2005; Dereli-İman, 2014; Doğru, M., Arslan, A., & Şeker, 2011; Kesicioğlu, 2015; Yıldırım, 2014).

Problem-solving skills are very important in supporting children's social development (Berk, 2013). Children with problem-solving skills are able to build healthy relationships with their friends, understand the emotions of those around them, and see events with other people's perspectives (Uysal, A. & Kaya-Balkan, 2015). Problem-solving skills are also very important in the temporary education system because they have a positive influence in instilling initial literacy and numeracy skills (Kalyuga, S., Renkl, A., & Paas, 2010).

Problem-solving is an important component of formal education programs at all levels, from preschool to higher education (Lazakidou & Retalis, 2010). This advanced cognitive skill can be described as the ability to solve a problem using rules and concepts (D. Wang et al., 2015). Educators hope that their pupils will leave their classes knowing how to solve problems. To improve these capabilities, they use open-ended questions and tasks (Woods et al., 1997). Students who participate in problem-solving exercises develop beneficial attitudes such as imagination, resilience, reasoning, and performance, which are important in real life.

Furthermore, both attitudes have been attributed to the ability to learn for the rest of one's life. As a result, challenges used in the teaching phase should be real-life situations that enable students to practice problem-solving skills (Yu et al., 2015). Many problem-solving models have been proposed to date. Sternberg's 7 Stages Approach is a well-known problem-solving model. Sternberg's seven-step approach can be applied to both

well-defined (known) and ill-defined (unknown) problems in which learners partake in a variety of epistemic beliefs (Lazakidou & Retalis, 2010).

The phases in this model are identifying the problem, defining the problem, developing a solution, organizing information, allocating resources, reporting, and assessing problem-solving. These phases of the problem-solving process necessitate simultaneous planning and decision-making. In addition, the learner should continue to practice and strengthen the problem-solving process to achieve the challenge (D. Wang et al., 2015). Individuals with intellectual disabilities can benefit from problem-solving skills in terms of independence and academic achievement (Erickson et al., 2015; Root et al., 2017).

Students with minor developmental difficulties will learn from multiple educational features by using schema-based learning in the practice method of teaching problem solving (Jitendra et al., 2013). As a result, visual representation is critical for special education pupils (Root et al., 2017). Any thinking techniques are used in problem-solving systems. Algorithmic and computational reasoning are the most general of these.

3 METHOD

This study used a class action research design using the Kemmis Taggart model (Kemmis et al., 2014), with two cycles covering, planning, action and observation, cycle reflection, planning, action, and reflection. The research was conducted in the odd semester of 2020/2021 and lasted for two months and ended after an increase in the quality of the process and learning outcomes in children's problem-solving skills. The research subjects were 20 children of Shafa Marwah's Play Group aged 5-6 years.

3.1 *Instrument*

Initial problem-solving skills is one of the fundamental aspects in early childhood cognitive development. Not only adults, early childhood also often encounter a problem. Therefore, problem solving is needed as a solution. Problem solving skills have their own stages according to their age. The instrument used to measure the initial problem-solving skills is the stage of the problem-solving pattern, including: (1) Understanding the problem (Un-understanding the problem); (2) Planning for problem solving (Devising a plan); (3) Implementing the completion plan (Currying out the plane); (4) Checking the results obtained (Looking back).

Table 1. Instrument Assessment of Problem-Solving Skills

Aspect	Indicator	Item	Amount
Understand the problem	Identify the content of the problem	1,2,3,4	4
	Identify the cause of the problem	5,6,7	3
Planning for problem solving	Found several solutions	8,9,10	3
	Predicting obstacles	11,12,13	3
Carry out the completion plan	Putting solutions into practice	14,15,16	3
Check the results obtained	Choose the best solution	17,18,19,20	4
		Total	20

3.2 *Procedure*

Researchers collaborate with class teachers, observe photo and video documentation of activities, when doing play activities, observations and interviews are carried out during home visits and do some stimulation through playing unplugged coding activities.

Before playing activities, the teacher makes plans by preparing various play unplugged activities, before starting the activity the teacher delivers Standard Operating Procedures (SOP) for playing activities to children, after the child understands the SOP given the teacher starts playing activities by inviting the child to choose the play activity that the child wants to do (1). Playing Lego, (2) Following the Pattern / Playing Origami (3). Puzzle Play. The teacher observes the child's initial problem-solving abilities in the process before, during and after playing activities. The teacher reflects on the problem-solving abilities of children who are still on an underdeveloped scale (UD) and are starting to develop (SD) to be given stimulation to develop according to expectations (DAE) and develop very well (DVW).

3.3 *Data Analysis*

Researchers analysed children's development outcomes in the initial problem-solving abilities of children aged 5-6 years using unplugged coding games. Researchers used qualitative and quantitative data analysis. Qualitative data analysis was obtained from the observation of photo / video documentation of children's activities and the results of interviews. Quantitative data analysis uses the calculation of the percentage increase in skala achievement in the development of children's initial problem-solving abilities before and after participating in unplugged coding activities.

4 RESULT AND DISCUSSION

4.1 *Result*

Pre-cycle data were obtained from observation, interviews, and documentation during pre-clusters. The initial assessment was carried out in August 2020. During the pre-cycle implementation, researchers encountered problems with the children's low initial problem-solving skills. One of the causes of children's low problem-solving abilities is the lack of variation in activities designed by the teacher. Teachers tend to provide regular play activities, so that children are less interested in participating in the learning process.

The learning implementation will be carried out for the initial problem-solving skills in cycle 1 will be carried out in September 2020. The results of the researchers' observations show that during play activities to implement unplugged coding as planned, there are still some children who must always be directed to find problem solutions when they encounter problems / obstacles when participating in playing activities. Researchers made field notes based on the results of observations and interviews which were used as assessment materials. Based on the results of data analysis (see table 2) in cycle 1 reached a percentage of 67.05% and increased again in cycle II to 80.05%. This proves that the child's initial problem-solving skills has increased. Unplugged coding is a play activity that can be used for the initial problem-solving skills of children aged 5-6 years.

In Cycle II, the average initial problem-solving skills of children increased by 70 points (in a percentage of 75.75%). When participating in playing activities, the children mentioned and explained the chronology of the problems faced, recognized the causes of the problems, mentioned some solutions and predicted consequences. Constraints of the solution to solve the problem, try to practice the solution to solve the problem, find the best solution and give reasons why the solution was chosen. The results of the analysis of this study indicate that unplugged coding can be applied to the initial problem-solving skills of children aged 5-6 years. This can be seen from 4 aspects of the 20 items that the researcher uses including aspects of Children being able to understand the problem (Ununderstanding the problem), planning for problem solving (Devising a plan), implementing the completion plan (Currying out the plane), examining the results in -Get (Looking back).

Table 2. Assessment of Problem-Solving Skills Results in Pre-cycle, Cycle I and Cycle II

No	Participants	Pre-cycle	Cycle I	Cycle II
1	SA	40	73	79
2	KA	40	71	82
3	NF	41	72	81
4	MA	39	60	63
5	MD	58	82	91
6	DV	40	60	83
7	DR	40	60	79
8	DK	39	66	84
9	SY	40	61	83
10	VI	41	62	65
11	MAK	40	60	78
12	AR	58	83	91
13	AMA	58	85	92
14	YSF	39	65	90
15	MB	39	61	62
16	DF	40	64	70
17	ARY	39	65	79
18	AUL	40	59	81
19	FN	41	63	84
20	SLS	40	69	83

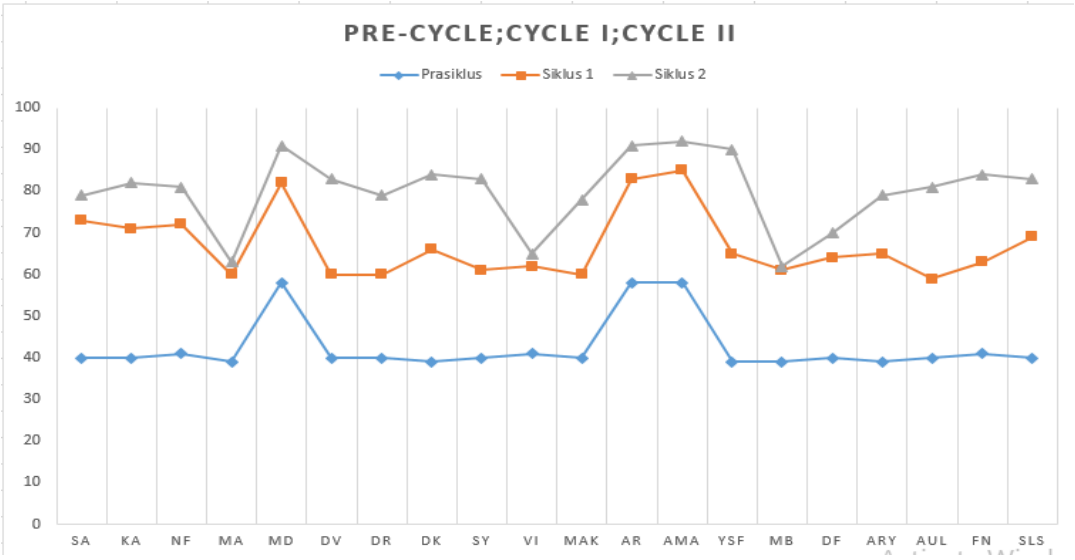

Figure 1. Graphs of Pre-cycle, Cycle I and Cycle II

Figure 1 shows the results of observations made by researchers for initial problem-solving skills starting from pre-cycle, cycle I and cycle II, as evidenced by the results of the second cycle of children showing better initial problem-solving skills compared to cycle I.

4.2 *Discussion*

The results of the analysis of this study indicate that unplugged coding can be applied to the early problem-solving skills of children aged 5-6 years. This is related to the difference in the learning methods of digital native children with ancient children. Children are more familiar with the service nowadays, and some also try to make their own (Pane, 2002). Concrete object activities will make it easier for kids to translate real environment logic into program logic, communicate with machines using physical objects, and participate in processes (McNerney, 2004). Most realistic data processing environments are much too difficult for children to perceive and comprehend. As a result, it is important that children learn to code without the use of machines. Computers are sophisticated and complicated to use, and their programs have many interconnected features that can be difficult for consumers to understand.

Programming for children aged 5-6 years must be connected to the real world and specific structures due to the complexity of programming (Montemayor et al., 2004). This can be seen from the findings of the observer during the unplugged coding activity with the children. This activity makes the children think carefully to find a way out. All challenges in problem solving programs through concrete objects make it easier for children to map problems and solutions.

The results also showed that activities with games involving concrete objects attracted children's interest. Game activities with the aim of honing the skill aspect of the problem require the teacher's efforts to update the level of the game more frequently, because children with an interest in unplugged coding activities can improve their skills quite quickly. This activity also involves a high degree of interaction between children. Preschool children can learn coding through realistic interactions, according to several scholars (Sullivan et al., 2015). Similarly, Wang et al., (2011) discovered that a curriculum focused on and combined with play activities was an engaging, useful, and simple-to-learn programming solution for kids. In contrast to conventional interventions, practices focused on concrete engagement, according to Horn et al., (2012), lead positively to learning.

Efforts to stimulate initial problem-solving abilities in early childhood must be done by providing interesting play activities. This is because using interesting play activities will make children excited about participating in the learning process. The implementation of research that has been carried out by researchers can show that in the learning

process, playing unplugged coding can stimulate initial problem solving abilities when children encounter problems, either during play activities or in being able to overcome potential problems in the future and develop them as needed (Anlıak, Ş., & Dinçer, 2005; Dereli-İman, 2014; Doğru, M., et al., 2011; Kesicioğlu, 2015; Yıldırım, 2014)

The activity of playing with unplugged coding is one of the most enjoyable play activities for children to stimulate their initial problem-solving skills optimally. Such as when participating in children's play activities mentioning and explaining the chronology of problems faced when playing Lego, Origami, puzzles, recognizing the causes of problems, mentioning some solutions, and predicting consequences / obstacles of solutions to solve problems, trying to practice solutions to solve problems, find the best solution and provide reasons why it was chosen. The activity of playing unplugged coding is not only for initial problem solving skills but also for children's social and social skills in terms of independence in self-directed learning (Threekunprapa, 2020).

The application of unplugged coding has a major impact on the initial problem-solving skills because children in the learning process while playing become more enthusiastic, excited, creative in pouring ideas and looking for the best solution when they encounter problems or problems while playing. The planning process is carried out optimally by the teacher by preparing several interesting play activities for children, and in the process of implementing teacher activities, it provides stimulation and motivation so that children are excited when participating in playing activities.

4.3 *Limitations*

During the research process, researchers have attempted to obtain complete data. However, due to time constraints, researchers were unable to document children's activities in the form of photos and videos.

5 CONCLUSION

The application of unplugged coding has an impact on initial problem-solving skills because using this program with a variety of play activities makes children enthusiastic, creative enthusiastic in pouring ideas, and looking for the best solution when they encounter problems or problems while playing. The process of implementing unplugged coding begins with the teacher preparing a variety of play activities in the form of Lego,

origami, and puzzles. The teacher provides stimulation and motivation when the child finds a problem by asking the child to name and explain the chronology of the problem at hand. Children can recognize the causes of problems, mention some solutions, and predict the consequences / obstacles of solutions to solve problems. Children are also able to try to practice solutions to solve problems, find the best solutions and provide reasons why these solutions were chosen by them. This unplugged coding program is an effective learning bridge to improve proficiency in other aspects of Computational Thinking.

6 REFERENCES

Akyol-Altun, C. (2018). Algorithm and coding education in pre-school teaching program integration the efectiveness of problem-solving skills in students.

Angeli, C., Smith, J., Zagami, J., Cox, M., Webb, M., Fluck, A., & Voogt, J. (2016). A K-6 Computational Thinking Curriculum Framework: Implications for Teacher Knowledge. Educational Technology & Society, 12.

Anlıak, Ş., & Dinçer, Ç. (2005). Farklı eğitim yaklaşımları uygulayan okul öncesi eğitim kurumlarına devam eden çocukların kişilerarası problem çözme becerilerinin değerlendirilmesi. Ankara Üniversitesi Eğitim Bilimleri Fakülte Dergis.

Aranda, G., & Ferguson, J. P. (2018). Unplugged Programming: The future of teaching computational thinking? Pedagogika, 68(3). https://doi.org/10.14712/23362189.2018.859

Arinchaya Threekunprapa. (2020). Patterns of Computational Thinking Development while Solving Unplugged Coding Activities Coupled with the 3S Approach for Self_Directed Learning. European Journal of Educational Research, 9(3), 1025–1045.

Arı, M. (2003). Türkiye'de erken çocukluk eğitimi ve kalitenin önemiNo Title. Erken Çocuklukta Gelişim ve Eğitimde Yeni Yaklaşımlar.

Armoni, M. (2012). Teaching CS in kindergarten: How early can the pipeline begin? ACM Inroads, 3(4), 18–19. https://doi.org/10.1145/2381083.2381091

Aydoğan, Y. (2004). İlköğretim ikinci ve dördüncü sınıf öğrencilerine genel problem çözme becerilerinin kazandırılmasında eğitimin etkisinin incelenmesi.

Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2009). Computer Science Unplugged: School students doing real computing without computers. 10.

Berk, L. E. (2013). Bebekler ve çocuklar: Doğum öncesinden orta çocukluğa. N. Işıkoğlu Erdoğan, Çev.

Bers, M. U. (2018). Coding, playgrounds, and literacy in early childhood education: The devel_opment of KIBO robotics and Scratch Jr. IEEE.

Brackmann, C. P., Moreno-León, J., Román-González, M., Casali, A., Robles, G., & Barone, D. (2017). Development of computational thinking skills through unplugged activities in primary school. ACM International Conference Proceeding Series, 65–72. https://doi.org/10.1145/3137065.3137069

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. 25.

Deek, F. P. (1999). The software process: A parallel approach through problem solving and program development. Computer Science Education.

DemiRer, V., & Sak, N. (2016). Programming Education and New Approaches Around the World and in Turkey. 26.

Dereli-İman. (2014). Değerler eğitimi programının 5-6 yaş çocukların sosyal gelişimine etkisi: Sosyal beceri, psiko-sosyal gelişim ve sosyal problem çözme becerisi. Kuram ve Uygulamada Eğitim Bilimleri.

Doğru, M., Arslan, A., & Şeker, F. (2011). Okul öncesinde uygulanan fen etkinliklerinin 5-6 yaş çocukların problem çözme becerilerine etkisi. Uluslararası Türkiye Eğiti Araştırmaları Kongresi.

Erickson, A. S. G., Noonan, P., Zheng, C., & Brussow, J. A. (2015). The relationship between self-determination and academic achievement for adolescents with intellectual disabilities. Research in Developmental Disabilities, 36, 45–54.

Fee, S. B., & Holland-Minkley, A. M. (2010). Teaching computer science through problems, not solutions. Computer Science Education, 20(2), 129–144. https://doi.org/10.1080/08993408.2010.486271

Futschek, G., & Moschitz, J. (2010). Developing algorithmic thinking by inventing and playing algo_rithms.

Gretter, S., & Yadav, A. (2016). Computational Thinking and Media & Information Literacy: An Integrated Approach to Teaching Twenty-First Century Skills.

Grover, S., & Pea, R. (2013). Computational thinking in k-12: A review of the state of the field. Educational Researcher.

Harrop, W. (2018). Coding for children and young adults in libraries: A practical guide for librarians. 45.

Hazzan, O., Lapidot, T., & Ragonis, N. (2011). Guide to Teaching Computer Science. Springer London. https://doi.org/10.1007/978-0-85729-443-2

Horn, M. S., Crouser, R. J., & Bers, M. U. (2012). Tangible interaction and learning: The case for a hybrid approach. Personal and Ubiquitous Computing, 16(4), 379–389. https://doi.org/10.1007/s00779-011-0404-2

Hsu, T.-C., Chang, S.-C., & Hung, Y.-T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. Computers & Education, 126, 296–310. https://doi.org/10.1016/j.compedu.2018.07.004

Ismail, M. N., Ngah, N. A., & Umar, I. N. (2010). Instructional strategy in the teaching of computer programming: A need assessment analyses. TOJET: The Turkish Online Journal of Educational Technology.

Ismail, M. N., Ngah, N. A., & Umar, I. N. (2010). Instructional Strategy in The Teaching of Computer Programming: A Need Assessment Analyses. The Turkish Online Journal of Educational Technology, 9(2), 7.

Jitendra, A. K., Petersen-Brown, S., Lein, A. E., Zaslofsky, A. F., Kunkel, A. K., Jung, P.-G., & Egan, A. M. (2013). Teaching Mathematical Word Problem Solving: The Quality of Evidence for Strategy Instruction Priming the Problem Structure. Journal of Learning Disabilities, 48(1), 51–72. https://doi.org/10.1177/0022219413487408

Joohi Lee. (2019). Coding in early childhood. Contemporary Issues in Early Childhood.

Kalyuga, S., Renkl, A., & Paas, F. (2010). Facilitating flexible problem solving: A cognitive load perspective. Educational Psychology Review.

Kemmis, S., McTaggart, R., & Nixon, R. (2014). The Action Research Planner. Springer Singapore. https://doi.org/10.1007/978-981-4560-67-2

Kesicioğlu, O. S. (2015). Okul öncesi dönem çocukların kişilerarası problem çözme becerilerinin incelenmesi. Eğitim ve Bilim.

Koksal Akyol, A. ve Didin, E. (2016). Ahlak gelisimi [Moral development]. In Cocuk Gelisimi icinde [In Child Development].

Lazakidou, G., & Retalis, S. (2010). Using computer supported collaborative learning strategies for helping students acquire self-regulated problem-solving skills in mathematics. Computers & Education, 54(1), 3–13. https://doi.org/10.1016/j.compedu.2009.02.020

Looi, C.-K., How, M.-L., Longkai, W., Seow, P., & Liu, L. (2018). Analysis of linkages between an unplugged activity and the development of computational thinking. Computer Science Education, 28(3), 255–279. https://doi.org/10.1080/08993408.2018.1533297

McClure, E. R., Guernsey, L., Clements, D. H., Bales, S. N., Nichols, J., Kendall-Taylor, N., & Levine, M. H. (2017). Grounding science, technology, engineering, and math education in early childhood. 68.

McLennan, D. P. (2017). Creating coding stories and games. Teaching Young Children.

McNerney, TimothyS. (2004). From turtles to Tangible Programming Bricks: Explorations in physical language design. Personal and Ubiquitous Computing, 8(5). https://doi.org/10.1007/s00779-004-0295-6

Mittermeir, R. T. (2013). Algorithmics for preschoolers—A contradiction?

Montemayor, J., Druin, A., Chipman, G., Farber, A., & Guha, M. L. (2004). Tools for children to create physical interactive storyrooms. Computers in Entertainment, 2(1), 12–12. https://doi.org/10.1145/973801.973821

Pane, J. F. (2002). A Programming System for Children that is Designed for Usability. 204.

Papanastasiou, G., Drigas, A., Skianis, C., Lytras, M., & Papanastasiou, E. (2018). Virtual and augmented reality effects on K-12, higher and tertiary education students' twenty-29 first century skills.

Pellegrino, J. W., & Hilton, M. L. (2012). Education for Life and Work: Developing Transferable Knowledge and Skills in the 21st Century.

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

Root, J., Saunders, A., Spooner, F., & Brosh, C. (2017). Teaching Personal Finance Mathematical Problem Solving to Individuals with Moderate Intellectual Disability. Career Development and Transition for Exceptional Individuals, 40(1), 5–14. https://doi.org/10.1177/2165143416681288

Scanlan, D. A. (1989). Structured flowcharts outperform pseudocode: An experimental comparison. IEEE Software, 6(5), 28–36. https://doi.org/10.1109/52.35587

Sheehan, K. J., Pila, S., Lauricella, A. R., & Wartella, E. A. (2019). Parent-child interaction and children's learning from a coding application. Computers & Education, 140, 103601. https://doi.org/10.1016/j.compedu.2019.103601

Shute, V. J., Sun, C., & Asbell-clarke, J. (2017). Demystifying computational thinking. Educational Research Review.

Sigelman, C. K., & Rider, E. A. (2012). Life-span Human Development (7th ed.). Cengage Learning.

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 Tech_nology and Design Education, 26, 3–20.

Sullivan, A. A., Bers, M. U., & Mihm, C. (2017). International conference on com_putational thinking education. Imagining, Playing, and Coding with KIBO: Using Robot_ics to Foster Computational Thinking in Young ChildreImagining, Playing, and Coding with KIBO: Using Robot_ics to Foster Computational Thinking in Young Children.

Sullivan, A., Elkin, M., & Bers, M. U. (2015). KIBO robot demo: Engaging young children in programming and engineering. Proceedings of the 14th International Conference on Interaction Design and Children, 418–421. https://doi.org/10.1145/2771839.2771868

Threekunprapa, A., & Yasri, P. (n. d.). (2020). The role of augmented reality based unplugged computer programming approach in the effectiveness of computational thinking.

Uysal, A. & Kaya-Balkan, İ. (2015). Sosyal beceri eğitimi alan ve almayan okul öncesi çocukların, sosyal beceri ve benlik kavramı düzeyleri açısından karşılaştırılması. Psikoloji Çalışmaları.

Vorderman, C. (2017). Computer coding for kids: A unique step-by-step visual guide, from binary code to building games.

Voronina, L. V., Sergeeva, N. N., & Utyumova, E. A. (2016). Development of algorithm skills in preschool children. Procedia-Social and Behavioral Sciences, 233, 155-159.

Wang, D., Han, H., Zhan, Z., Xu, J., Liu, Q., & Ren, G. (2015). A problem solving oriented intelligent tutoring system to improve students' acquisition of basic computer skills. Comput. Educ., 81, 102–112.

Wang, D., Zhang, C., & Wang, H. (2010). Proceedings of the 10th international conference on interaction design and children. T-Maze: A Tangible Programming Tool for Children.

Wang, Danli, Zhang, C., & Wang, H. (2011). T-Maze: A tangible programming tool for children. Proceedings of the 10th International Conference on Interaction Design and Children - IDC '11, 127–135. https://doi.org/10.1145/1999030.1999045

Woods, D. R., Hrymak, A. N., Marshall, R. R., Wood, P. E., Crowe, C. M., Hoffman, T. W., Wright, J. D., Taylor, P. A., Woodhouse, K. A., & Bouchard, C. G. K. (1997). Developing Problem Solving Skills: The McMaster Problem Solving Program. Journal of Engineering Education, 86(2), 75–91. https://doi.org/10.1002/j.2168-9830.1997.tb00270.x

Yıldırım, A. (2014). Okul öncesinde yaratıcı problem çözme etkinliklerinin yaratıcılığa etkisi (5 yaş örneği). Hacettepe University, Ankara, Turkey.

Yohanes. (2018). Mengajarkan Computational Thinking dan Coding Pada Anak-Anak. Amazing Grace. https://blog.compactbyte.com/2018/05/26/mengajarkan-computational-thinking-dan-coding-pada-anak-anak/

Yu, K.-C., Fan, S.-C., & Lin, K.-Y. (2015). Enhancing Students' Problem-Solving Skills Through Context-Based Learning. International Journal of Science and Mathematics Education, 13(6), 1377–1401. https://doi.org/10.1007/s10763-014-9567-4

Yuksel, H. S. (2019). Experiences of Prospective Physical Education Teachers on Active Gaming within the Context of School-Based Physical Activity. European Journal of Educational Research, 8(1). https://doi.org/10.12973/eu-jer.8.1.199

Zvarych, I., Kalaur, S. M., Prymachenko, N. M., Romashchenko, I. V., & Romanyshyna, O. Ia. (2019). Gamification as a Tool for Stimulating the Educational Activity of Students of Higher Educational Institutions of Ukraine and the United States. European Journal of Educational Research, 8(3). https://doi.org/10.12973/eu-jer.8.3.875