Mathematics Preservice Teachers' Preparation in Designing Mathematics-Based Programming Activities Rich in Metacognitive Skills

https://doi.org/10.3991/ijet.v18i06.36965

Wajeeh Daher[1,2], Nimer Baya'a[1(✉)], Otman Jaber[1]
[1]Al-Qasemi Academic College of Education, Baqa-El-Gharbia, Israel
[2]An-Najah National University, Nablus, Palestine
bayaan@qsm.ac.il

Abstract—Programming problems enrich the environment of mathematics learning, adding the flavor of technology to these problems. This is especially true when this programming is Scratch based, where Scratch is being used to make students' learning of mathematics more meaningful. This role of programming in the mathematics classroom points at the importance of preparing mathematics teachers for designing mathematics-based programming problems activities. The present research describes one attempt to prepare mathematics preservice teachers in designing mathematics-based programming problems activities that could be used in the classroom to teach both programming and mathematics concepts. Twenty-three preservice teachers participated in the research, where they worked in eight groups of 2–3 members in each group. Data were collected through observations based on video recordings of the sessions in which the preservice teachers discussed with the pedagogical supervisors the designed mathematics activities. The preparation model comprises of five stages related to the educational environment and to the design notions. The results show special importance for the concepts of struggle and devolution in designing this kind of activities, in addition to the concept of equilibrium between the creative and imitative thinking. The results also show the useful application of metacognitive skills when designing the activities, especially when designing the directions given to the students for solving each of the programming activities.

Keywords—activity design, metacognitive skills, Scratch environment, algorithmic and creative processes, struggle and devolution processes

1 Introduction

Researchers have been interested in mathematics-based programming activities as it could support students' construction of mathematical ideas (e.g., [1]). More emphasis has been put on solving mathematics-based programming activities than on designing them. The present study intends to add to the research on designing programming activities based on mathematical ideas. It studies preservice teachers' (PST) design of such activities in the frame of their third-year preparation as mathematics teachers in a college of education. Coming to design such activities, a key part of the PSTs was to facilitate students' metacognition. Further, the design processes were enabled emphasizing design concepts of algorithmic, creative, struggle and devolution processes. These concepts are part of a theoretical framework suggested by Lithner [2–3]. Specifically, the present paper is interested in the preparation phases of the PSTs to design mathematics-based Scratch-programming activities that encourage the metacognitive processes of students.

1.1 Literature review

Researchers considered metacognition as knowledge about knowledge or cognition about cognition [4–6]. According to Flavell [4], metacognition is the process of an individual consciously aware of, considering, and controlling cognitive processes and strategies. Various interpretations of the term have developed since then, according to Du Toit and Kotze [7], where several definitions of metacognitive processes, including Schoenfeld's [8] definition, attribute importance to monitoring and regulating cognitive processes. Metacognition was defined by Flavell [9] as knowledge that adapts the cognitive processes implicated in an activity so that it promotes effective understanding. Gavelek and Raphael [10] have argued that metacognition involves adjusting the cognitive processes involved in the process to enhance understanding (p. 8). A study by Panaoura et al. [5] says that metacognition affects several variables of students' learning, especially cognition, so it impacts their academic success.

Mathematics education researchers have been studying metacognitive skills. In a study by Barbacena and Sy [11], metacognition awareness, metacognition evaluation, and metacognition regulation were found to be pathways between metacognitive skills in university students' problem solving. Daher et al. [12] found that metacognitive processes help maintain a positive social and emotional climate in the mathematics classroom.

1.2 Scratch in problem solving

In Scratch, users program by sequencing code blocks to create programs, which are identified by researchers as block-based visual programming languages. Scratch programming activities can be effectively used for the consideration of mathematics when using block-based programming as an educational tool [13–14]. In a study conducted by Taylor et al. [15], when learning mathematics, students used Scratch to develop strategies of cooperative learning and to establish goals, generate ideas, and test them. According to Rodriguez-Martinez et al. [16], students in sixth grade can strengthen their mathematical thinking and computational skills by using the Scratch programming language. In addition, Calder [17] examined the emergence of mathematical thinking among children who use Scratch, which demonstrated that Scratch positively affects children's motivation and engagement in learning mathematical concepts through their use of the programming language. The ministry of education in Israel issued a Scratch curriculum for computer science in the primary and in the middle schools [18]. The curriculum in the primary school is called 'Computer science and robotics', while the curriculum in the middle school is called 'Algorithmics with Scratch Environment'.

1.3 Theoretical framework

Davidson and Sternberg [19] suggested a framework that includes a sequence of metacognitive skills: encoding, representation, decomposition, planning, selecting strategy, monitoring, evaluating, and looking for other strategies. In the current research, we mainly considered design issues, by exploiting the aforementioned framework to describe the metacognitive processes to mathematics PSTs. Exploiting Davidson and Sternberg framework, we operationalized the metacognitive processes in the following manner:

Encoding: The main words and symbols in the problem should be identified and written.

Representation of the givens: Representing the givens algebraically; through equations for example, or visually; through a graph for example.

Problem decomposition: Decomposing the problem into pieces and a series of procedures.

Planning: Writing a series of actions for a solution method.

Strategy selection and implementation: Examining all possible solutions or sequences of solutions to solve the problem.

Plan monitoring: Evaluating the progress of the solution process in terms of achieving the goal.

Solution evaluation: Assessing the correctness of each solution procedure.

Looking for other solutions: Evaluating whether other solutions are applicable to the problem.

Assessing the used strategies: Evaluating whether the used strategies supported the arrival at the solution in an effective way.

Looking for other strategies: Searching for other strategies that would improve effectively the solution process.

A second theoretical framework on which the activity-design preparation depends is that of algorithmic, creative, struggle and devolution processes. This theoretical framework suggested to adopt the framework of creative reasoning and imitative reasoning by Lithner [2–3], where Jonsson et al. [20] called the imitative reasoning by the algorithmic reasoning. This framework allows for mathematical "struggle" in ad hoc situations (with no assistance from the teacher), where students construct their solutions independently. In the present research, the PSTs are trained to design activities, which present programming problems that students must "struggle", in a productive sense, with programming, mathematical knowledge and thinking while solving them, "… but a delicate balance must be struck to prevent these struggles from becoming obstacles rather than promoters of learning" [2]. The productive struggle requires active engagement in non-routine programming problem solving rather than carrying out algorithmic solution processes or imitating given solution templates.

According to Artigue ([21], p. 160), it's critical to pay attention to the features of the learning context in which the learners interact to allow autonomous action and productive feedback, which indicates that students' autonomy is one of the factors we need to consider when designing activities, where the decision making of the student is another factor that we should manage. This autonomy is subject to the devolution level that the designer considers when writing directions for the students in the activity. Learners take responsibility for a portion of the process of problem-solving. Therefore, the directions given in the activity should balance between systematic solution processes, and general non-useful directions. Throughout the process of accepting the problem and producing a solution, the directions are to support the students, but not to suggest a solution for them. To achieve this balance, the PSTs were introduced to the metacognitive skills as an outline for writing suitable directions in the activities to support the students in their struggle for solving the programming problems with proper level of devolution.

1.4 Research rationale and goals

Research on metacognition in Scratch has been limited. Cho et al. [22] described a study in which Scratch was used successfully to develop a game that significantly improved middle school students' metacognition. Daher et al. [1] reported that PSTs can develop meta-cognitive processes in learning and teaching using a programming environment. Daher et al. [23] considered PSTs' preparation in solving and designing mathematical activities that emphasize metacognitive processes but without emphasis on programming. They recommended that during this education, PSTs should solve challenging problems emphasizing metacognitive skills, teach by using such problems, and design them. Throughout the whole process, the PSTs should discuss their ways, and consider the whole process of engaging in metacognition.

In addition to the above, students in the middle school study algorithmics with the Scratch environment, which points at the need of PSTs to learn how to design programming activities in this environment. In the present research, we attempt to investigate the appropriate trajectory of developing the design processes of PSTs who are concerned with metacognitive processes in mathematics-based programming problems by using the Scratch language.

2 Methodology

2.1 Research context and participants

The study was performed in a teachers' seminar among 23 third-year middle-school PSTs majoring in mathematics and computers. All the PSTs were expected to participate in the study in the frame of their B.Ed. study in the seminar. The learning context was that of practical training in the frame of "class academy" project in a middle school, where we utilized this context to prepare our PSTs in designing mathematics-based programming problems activities that could be used in the classroom to teach both programming and mathematics concepts. During each day of practical training in the school, the authors were responsible for implementing the preparation process in a workshop. These authors served as pedagogical supervisors for the PSTs in the middle school. The PSTs had previous knowledge of the Scratch environment, as part of their study in the second year, and they worked in eight groups of 2–3 members.

To elaborate, we asked each group of the PSTs to design an activity involving a programming problem in Scratch, based on mathematical knowledge. The PSTs were requested to explain their design considerations. They were asked to discuss these considerations in Zoom meetings and upload the recorded meetings to the Moodle site of the practical training course. Each group was requested to upload files which include the following content: the final design of the activity, considerations of choosing the programming problem, the mathematical knowledge needed to solve the problem and the directions given to the students for performing the activity. The PST uploaded also Zoom recording of the meetings, procedures and steps needed for solving the programming problem, the code in Scratch suggested as a solution for the programming problem and an evaluation of their consideration of the effectiveness in designing the activity. The previous materials were the base for the workshop discussions in the five preparation phases that are described in the results.

2.2 Data collecting and analysis tools

Data were collected through observations based on video-recordings of the workshop sessions in which the PSTs discussed with the pedagogical supervisors the designed mathematics-based programming activities. We analyzed the data deductively and inductively according to the design components. The design components are Scratch programming, metacognitive skills, and design processes. The design processes are algorithmic processes, creative processes, struggle processes and devolution processes.

3 Results

3.1 First stage: intuitive design

In the first stage, before designing the first activity, the pedagogical supervisors (called supervisors below for abbreviation) concentrated on PSTs' intuitive design of the activities. Transcript 1 illustrates the discussion between the supervisors and the PSTs in the first stage.

11	Supervisor 1:	You are requested to design mathematics activities for the middle school. These activities need to fit working in the Scratch environment. You can meet at the beginning to discuss issues related to this design. You do not need, at the first meeting, to decide upon the specifics of the activity.
12	PST 1	Why do not we decide upon our design approach at the first meeting?
13	Supervisor 2:	You can. You decide for yourself how to go on designing the activities.
14	Supervisor 3:	We are talking about imitative processes and creative ones. Creative processes need an incubation process to evolve. This is according to a framework that describes the development of creative processes.
15	PST 2:	O.K. We will decide upon our design, taking into account this issue of the incubation needed to develop our creative design processes.

Transcript 1. Discussing design processes

Supervisor 1 sets the stage for the intuitive design, where he reminds the PSTs that they need not to decide on the specifics of the activity [R11]. Supervisor 2 supports this way of thinking by emphasizing the freedom of the PST to decide upon the design method [R13]. Supervisor 3 lessens somehow this intuitive design by talking about design notions, i.e., imitative, and creative processes. Nonetheless, the PSTs understand that they are the main players in this design [R15].

3.2 Second stage: struggle and devolution design

In the second stage, before designing the second activity, the supervisors introduced the concepts of struggle and devolution while designing activities that promote algorithmic/creative problem-solving processes. Transcript 2 illustrates the classroom discussion during this stage.

111	Supervisor 1:	The paper says that the activity should include struggle, which means that the problem should be in some place between too straight forward and too hard. The student shouldn't struggle too much, as great struggle could make her or him lose interest in solving the problem. She or he will also lose interest if she or he can solve the problem without any struggle.
112	PST 3	How can we do that?
113	Supervisor 1	You should decide when you meet as a group.
114	Supervisor 2:	Coming to design the activity, you should take into consideration that the problem should be a programming problem that requires mathematical knowledge to develop the algorithmic solution of the problem. In addition, the problem should not be a mathematical problem that would be solved on paper, and then validate the solution by utilizing the Scratch environment.
115	Supervisor 3:	Continuing what Supervisor 1 has described, we have also the notion of devolution in which the teacher should give responsibilities for the students in solving the problem. The directions should not get to the level of algorithmic step by step instructions for the solution process. At the same time, the directions should not be very general leaving all the solution process in the responsibility of the student. One solution could be approached by using the metacognitive skills, but we need to adjust them to the content of the problem.
116	PST 4:	Is devolution what we call investigative activities?
117	Supervisor 3:	In a way yes, but it also includes giving the students power over their learning.

Transcript 2. Discussing the algorithmic/creative based design process

Supervisor 1 starts the introduction of the concepts of struggle and devolution, by describing the concept of struggle [R111]. Supervisor [R113] 1 does not answer the PST3's question directly and thus emphasized this concept of struggle. Supervisor 3 introduces the notion of devolution, by emphasizing the responsibility of the student in the solution process.

3.3 Third stage: metacognitive design

In the third stage before designing the third activity, the supervisors introduced the PSTs to the metacognitive skills of Davidson and Steinberg [19], recommending using them in giving directions to the students while solving the problems. Transcript 3 illustrates the beginning of the class discussion, where we will not describe all the discussion as it is very long.

216	Supervisor 2:	Here is the problem that we want to discuss its solution: A new teacher has been appointed to a school in a neighbouring city. She wants to choose the best way to arrive at the school. How could she decide upon that?
217	PST 5:	We need first to write down the givens of the problem: the city from which the teacher will depart, the city to which the teacher will arrive, all roads connecting the first city to the second, etc.
218	Supervisor 1	So, what do you suggest?
219	PST 5:	To write down the givens of the problem.
220	Supervisor 1	O.K. What is the next step?

Transcript 3. Discussing the integration of metacognitive skills in activity design

Supervisor 2 starts introducing the metacognitive skills of Davidson and Steinberg [19], by giving the PSTs a problem to solve, and asking about the procedure of solving it [R216]. This makes the PSTs to suggest the first metacognitive skill, i.e., writing the givens [R217].

3.4 Fourth stage: metacognitive and algorithmic/creative design

In the fourth stage, before designing the fourth activity, the supervisors discussed the third activity and concentrated on the adjustment of the metacognitive skills to the problem. While designing the directions given to the students, the supervisors and prospective teachers discussed the balance between the algorithmic and creative aspects of the given directions.

321	PST 6:	The problem that we wrote is about a Parking lot in a school, where the teachers have a daily problem of parking their cars in the park. We requested the student to write a program that computes the number of cars that the park could include.
322	Supervisor 2:	Is the problem about a specific park?
323	PST 7:	No. It is about a general park.
324	Supervisor 2:	What is the figure of the park.
325	PST 7:	It is a rectangle.
326	Supervisor 2:	Where is that written?
327	PST 6:	We suppose that this is known.
328	Supervisor 2:	We cannot assume that. How does this problem encourage metacognitive processes?
329	PST 8:	We, as teachers, do that orally.
330	Supervisor 2:	We need to do that through the instructions of the problem. Think about that before you design your next problem.
331	Supervisor 1:	How do you encourage the creativity of students in solving the problem?
332	PST 7:	We need to request them to approach the problem in different ways.
333	Supervisor 3:	What are you referring to? To the mathematical solution or to the programming solution?
334	PST 8:	We did not consider that. [She directed her speech to her group's members] We also need to consider this issue when we design our next problem.

Transcript 4. Discussing the design of metacognitive skills and algorithmic/creative processes

Supervisor 2 emphasizes the metacognitive skill of considering the givens [R328]. Supervisor 3 discusses the creativity issue of the activity, asking about the way to do that [R331]. In addition, supervisor 3 emphasized that the issue of creative solutions could refer to the mathematical as well as the programming aspects [R333].

3.5 Fifth stage: reflection on design components

In the fifth stage after designing the fourth activity, the supervisors, together with the PSTs, reflected on the different design components in the fourth activity, especially students' devolution and their struggle in solving the problem. Doing that, the PSTs and supervisors reflected on the components of design. The supervisors emphasized the need for balance in the algorithmic vs creative aspects, in the directions of the programming activity using the metacognitive skills properly.

417	Supervisor 2:	Can you please present your programming problem?
418	PST 11:	Of course. Here it is. Write a Scratch program that receives the length of a segment, draws it horizontally to the right starting with the point (0,0). Then, it receives the distance of a point from the middle of the segment, and then draws a triangle whose points are the vertices of the segment and the drawn point. Two sprites will then appear at the point (0,0) and start to move keeping the same speed. The first sprite moves along the segment, while the second sprite moves from one vertex of the segment to the other through the drawn point. You need to run the program several times and write mathematical conclusions. You can watch first the accompanying video that demonstrates the required outcome.
419	Supervisor 1:	Can you please show us the activity directions to the students?
420	Group:	[The group of PSTs shows the directions given to the students in the different phases of metacognition].
421	PST 11	Here are the directions to the students: *Encoding the givens*: 1. What did you see when watching the video? 2. What are the givens, according to the video? 3. Consider whether there are incomplete or redundant givens in the problem.
422	Supervisor 2	[Talking to the class] What do you think about the encoding?
423	PST 11	They are not many, nor few.
424	Supervisor 2	Why did you need the first direction?
425	PST 12	Probably we can give it up.
426	PST 11	*Representation*: 4. How can we represent the sprites' movements? 5. What are the mathematical objects and processes needed in the representation?
427	Supervisor 3	[Talking to the class] What do you think about the representation?
428	PST 13	We wrote two questions, one for the programming representation and one for the mathematical representation.
429	PST 11	*Problem decomposition*: 6. How can we program the appearances and movements of the different sprites?
430	Supervisor 2	[Talking to the class] What do you think about the decomposition?
431	PST 13	Here, we thought that the difficulty might be a programming one, so we asked about it.
432	Supervisor 1	You need to think about a mathematical difficulty in the decomposition.

Transcript 5. Discussing the devolution and struggle processes

The supervisors, along the transcript, ask the PSTs questions to verify metacognitive skills in the activity directions and the struggle and devolution issues. The main question there is 'What do you think about the given directions?'. Here this question is in terms of struggle and devolution. Supervisor 2 addresses the issue of redundancy of the directions, where this redundancy could lead to less sufficient struggle [R424]. Supervisor 1 addresses the issue of required directions, where this issue could lead to more than needed struggle [R432].

4 Discussion

The present paper aimed to show a trajectory for PSTs' preparation in designing mathematics-based programming activities. It emphasizes the sequence of stages of this preparation and the goals of each stage in this sequence. In the first stage before designing the first activity, the concentration should be put on PSTs' intuitive design of activities, where researchers have put emphasis on intuitive thinking especially in design (e.g., [24]). In the second stage, after designing the first activity and before designing the second activity, discussion in the whole class should be held regarding the components of design, especially the concepts of struggle and devolution while designing activities that promote algorithmic/creative problem-solving processes. Here, struggle should be introduced as the equilibrium between too hard and straight forward problem solving, so the students would not find the struggle too easy and get fed up or too hard and loose interest. This struggle needs to lead to equilibrium between the algorithmic and the creative [2]. As the problem needs to fit the programming environment, it should not be a mathematical problem that would be solved on paper, and then just present the solution in the Scratch environment. In this stage, devolution needs to be in students' hands, where the design needs to give responsibilities for the students in solving the problem. The directions should not get to the level of algorithmic step by step instructions for the solution process and should not be very general leaving all the solution process in the responsibility of the students [25].

In the third stage, after design the second activity, and before designing the third activity, the concentration should be focused on the previous issues of struggle and devolution, but also on the introduction of metacognitive concepts. Here, we utilized the meta-cognitive skills of Davidson and Steinberg [19] and recommended using them in the directions to the students. Other frameworks could be used as that of Wilson and Clarke [26]. Emphasis should be put on scaffolding, self and other scaffolding, and individual and collective scaffolding. This emphasis on scaffolding as design element encourages the ability of learners to have control over their learning process to a certain extent by relying on the support offered by teachers and the self [27].

In the fourth stage, after designing the third activity and before designing the fourth activity, the concentration should be on the adjustment of the meta-cognitive skills to the nature of the problem. Here, we need to address specifically the blending of meta-cognitive processes and struggle and devolution processes. Teachers find this blending difficult at the beginning, but with continuous experimentation and scaffolding, they could proceed towards successful design [28].

In the fifth stage, concentration should be on how to consider the combination of all the design issues. Special focus should be on the struggle and devolution processes and how these issues are connected to the algorithmic and creative processes [4]. Appropriate levels of struggle and devolution could lead to equilibrium between the algorithmic and creative processes and lead to creative products.

5 Conclusions recommendations and limitations

Researchers have been interested in digital learning since the advent of digital tools since it influences positively students' learning [29–32]. One such aspect of digital learning is the use of Scratch-based programming in mathematics education. The present research came to investigate how preservice teachers could be prepared for designing mathematics-based activities using the Scratch programming, when the emphasis is put on metacognitive activities. The findings suggest five stages of preparation. These phases take care of the various components of the design, especially metacognitive skills, Scratch programming and design processes, where the design processes were mainly algorithmic/creative processes and struggle/devolution processes. It is recommended that teacher colleges prepare curricula that prepares preservice and in-service teachers for designing subject-based programming activities, not only for the mathematics classroom but for other disciplines. The sequence of stages for the preparation suggested in the present research could be an effective trajectory to begin with, but modifications could be applied according to the discipline and programming language.

One limitation of the study is that it was implemented in one teacher college and one group of students. Further implementations and accompanying studies are needed to verify the fitness of the suggested trajectory and its stages.

Another limitation of the study is that it utilized specific models of metacognition and design processes. Attempts could be done using other models.

6 References

[1] Daher W, Baya'a N, Jaber O, Awawdeh Shahbari J. (2020). A Trajectory for Advancing the Meta-Cognitive Solving of Mathematics-Based Programming Problems with Scratch. *Symmetry, 12*(10), 1627. https://doi.org/10.3390/sym12101627

[2] Lithner, J. (2008). A research framework for creative and imitative reasoning. *Educational Studies in Mathematics, 67*(3), 255–276. https://doi.org/10.1007/s10649-007-9104-2

[3] Lithner, J. (2017). Principles for designing mathematical tasks that enhance imitative and creative reasoning. *ZDM Mathematics Education, 49*, 937–949. https://doi.org/10.1007/s11858-017-0867-3

[4] Flavell, J. (1976). Metacognitive aspects of problem solving. In L. Resnick, (Ed), *The Nature of Intelligence* (pp. 231–235). Erlbaum Associates: New Jersey.

[5] Panaoura, A., Philippou, G., & Christou, C. (2003). *Young pupils' metacognitive ability in mathematics*. Paper presented at the Third Conference of the European Society for Research in Mathematics Education.

[6] Veenman, M. V. J. (2005). The assessment of metacognitive skills: What can be learned from multimethod designs? In C. Artelt, & B. Moschner (Eds), *Lernstrategien und metakognition: Implikationen fur forschung und praxis* (pp. 75–97). Berlin: Waxmann.

[7] Du Toit, S., & Kotze, G. (2009). Metacognitive strategies in the teaching and learning of mathematics. *Pythagoras,* 70, 57–67. https://doi.org/10.4102/pythagoras.v0i70.39

[8] Schoenfeld, H. (1992). Learning to think mathematically: Problem solving, metacognition and sense making in mathematics. In D. A. Grouws (Ed), *Handbook of research on mathematics teaching and learning* (pp. 334–368). New York: McMillan.

[9] Flavell, J. H. (1979). Metacognition and cognitive monitoring: A new area of cognitive–developmental inquiry. *American Psychologist, 34*, 906–911. https://doi.org/10.1037/0003-066X.34.10.906

[10] Gavelek, J. R., Raphael, T. E. (1985). Metacognition, instruction, and questioning, In D. L. Forrest-Pressley, G. E. MacKinnon, T. G. Waller (Eds.), *Metacognition, cognition, and human performance* (Vol. II, pp. 103–132), Orlando: Academic Press. https://doi.org/10.1016/B978-0-12-262302-8.50008-0

[11] Barbacena, L., & Sy, N. (2015). Metacognitive model in mathematical problem solving. *Intersection, 12*(1), 16–22. https://doi.org/10.1016/j.learninstruc.2005.12.003

[12] Daher, W., Anabousy, A., & Jabarin, R. (2018). Metacognition, positioning and emotions in mathematical activities. *International Journal of Research in Education and Science (IJRES), 4*(1), 292–303. https://doi.org/10.21890/ijres.383184

[13] Grover, S., Pea, R., & Cooper, S. (2015). Designing for deeper learning in a blended computer science course for middle school students. *Computer Science Education, 25*(2), 199–237. https://doi.org/10.1080/08993408.2015.1033142

[14] Han, B., Bae, Y., & Park, J. (2016). The effect of mathematics achievement variables on Scratch programming activities of elementary school students. *International Journal of Software Engineering and Its Applications, 10*(12), 21–30. https://doi.org/10.14257/ijseia.2016.10.12.03

[15] Taylor, M., Harlow, A., & Forret, M. (2010). Using a computer programming environment and an interactive whiteboard to investigate some mathematical thinking. *Procedia-Social and Behavioral Sciences, 8*, 561–570. https://doi.org/10.1016/j.sbspro.2010.12.078

[16] Rodríguez-Martínez, J. A., González-Calero, J. A., & Sáez-López, J. M. (2020). Computational thinking and mathematics using Scratch: An experiment with sixth-grade students. *Interactive Learning Environments, 28*(3), 316–327. https://doi.org/10.1080/10494820.2019.1612448

[17] Calder, N. (2010). Using Scratch: An integrated problem-solving approach to mathematical thinking. *Australian Primary Mathematics Classroom, 15*(4), 9–14.

[18] Ministry of Education (2012, June). Computer science major, software engineering and cyber. https://edu.gov.il/tech/MadaTech/megamot/Pages/csit.aspx

[19] Davidson, J. E., & Sternberg, R. J. (1998). Smart problem solving: How metacognition helps. *Metacognition in Educational Theory and Practice,* 47–68.

[20] Jonsson, B., Norqvist, M., Liljekvist, Y., & Lithner, J. (2014). Learning mathematics through algorithmic and creative reasoning. *The Journal of Mathematical Behavior, 36*, 20–32. https://doi.org/10.1016/j.jmathb.2014.08.003

[21] Artigue, M. (2015). Perspectives on design research: The case of didactical engineering. In *Approaches to qualitative research in mathematics education* (pp. 467–496). Springer, Dordrecht. https://doi.org/10.1007/978-94-017-9181-6_17

[22] Cho, S. H., Song, J. B., Kim, S. S., & Paik, S. H. (2008). The effect of a programming class using Scratch. *Journal of the Korean Association of Information Education, 12*(4), 375–384.

[23] Daher, W., Bayaa, N., Jaber, O., & Anabousy, A. (2018). Developing pre-service mathematics teachers' metacognitive thinking for learning and teaching with mobile technology. The Eurasia Proceedings of Educational and Social Sciences, 9, 239–249.

[24] Panbanlame, K., Sangaroon, K., & Inprasitha, M. (2014). Students' intuition in mathematics class using lesson study and open approach. *Psychology, 5*(13), 1503. https://doi.org/10.4236/psych.2014.513161

[25] Daher, W., Abo Mokh, A., Shayeb, S., Jaber, R., Saqer, K., Dawood, I., Bsharat, M., & Rabbaa, M. (2022). The design of tasks to suit distance learning in emergency education. *Sustainability, 14*(3), 1070. https://doi.org/10.3390/su14031070

[26] Wilson, J., & Clarke, D. (2004). Towards the modelling of mathematical metacognition. *Mathematics Education Research Journal, 16*(2), 25–48. https://doi.org/10.1007/BF03217394

[27] Holton, D., & Clarke, D. (2006). Scaffolding and metacognition. *International Journal of Mathematical Education in Science and Technology, 37*(2), 127–143. https://doi.org/10.1080/00207390500285818
[28] Daher, W., Baya'a, N., & Jaber, O. (2022). Understanding prospective teachers' task design considerations through the lens of the theory of didactical situations. *Mathematics, 10*(3), *417*. https://doi.org/10.3390/math10030417
[29] Abuzant, M., Ghanem, M., Abd-Rabo, A., & Daher, W. (2021). Quality of using Google classroom to support the learning processes in the automation and programming course. *International Journal of Emerging Technologies in Learning (iJET), 16*(06), 72–87. https://doi.org/10.3991/ijet.v16i06.18847
[30] Daher, W., & Awawdeh Shahbari, J. (2020). Design of STEM activities: Experiences and perceptions of prospective secondary school teachers. *International Journal of Emerging Technologies in Learning (iJET), 15*(04), 112–128. https://doi.org/10.3991/ijet.v15i04.11689
[31] Al Breiki, M., & Al-Abri, A. (2022). The extended technology acceptance model (ETAM): Examining students' acceptance of online learning during COVID-19 pandemic. *International Journal of Emerging Technologies in Learning (iJET), 17*(20), 4–19. https://doi.org/10.3991/ijet.v17i20.29441
[32] Georgieva-Tsaneva, G. N., & Serbezova, I. (2022). Research on the impact of innovative interactive technologies in the education of health care students. *International Journal of Emerging Technologies in Learning (iJET), 17*(20), 283–291. https://doi.org/10.3991/ijet.v17i20.32903

7 Authors

Wajeeh Daher is a full professor in the educational sciences department at An-Najah National University and head of mathematics department at Al-Qasemi Academic College of Education. A significant part of his research focuses on the different aspects of using technology in education. One main issue focuses on the professional development of pre-service and in-service mathematics teachers in using digital tools to encourage school students' high order thinking, creative thinking, critical thinking and metacognitive thinking (wajeehdaher@gmail.com).

Nimer Baya'a is a senior lecturer A in the mathematics education program of the graduate studies, pedagogical supervisor, and head of the ICT center at "Al-Qasemi" Academic College of Education – Baqa El-Gharbiah. He was also for 25 years the supervisor of computers in the Arab schools in the Ministry of Education in Israel. His research focuses on using ICT, digital tools, social networks, cellular phones, and programming in Scratch in teaching mathematics. Currently, concentrates on professional development of pre-service mathematics teachers in assimilating meta-cognition, HOTS, and creative thinking among their students (bayaan@qsm.ac.il).

Otman Jaber is a lecturer and pedagogic supervisor at Al-Qasemi Academic College of Education. His field of interest focuses on mathematics, professional development of pre-service teachers. Recently, his research focuses on investigating the potential of the Augmented-Reality (AR) technology in learning mathematical concepts (Zeta275@yahoo.com).

Article submitted 2022-11-21. Resubmitted 2023-01-14. Final acceptance 2023-01-14. Final version published as submitted by the authors.