Session 2: *IT Education and Practice*
Original paper
UDC: 371.214:004

Qualitative Analysis of Problem-Solving Process in the Field of Programming in Primary Education

Biljana Đorić [1*], Dragan Lambić [2]
[1] University of Kragujevac, Faculty of Technical Sciences, Čačak, Serbia
[2] University of Novi Sad, Faculty of Education, Sombor, Serbia
[*] biljana.djoric@ftn.kg.ac.rs

Abstract: *Digital competencies are necessary for the functioning of an individual in contemporary society and therefore it is important to develop the same in students through whole elementary education. The skills of algorithmic, computing and logical thinking are the essential part of these competencies. In order to plan curriculum content in the field of programming well, it is important to understand how students see the very process of problem solving .Therefore, the aim of this work is to identify categories of fourth grade students' behavior while dealing with problem tasks in the field of programming. Twenty-nine fourth grade students took part in this research. The technique of collecting data using video recordings of the problem-solving process was applied. Qualitative analysis of data has shown categories of behavior which might arise when dealing with solving tasks of different content and complexity. Besides behavior, the analysis of video recordings has pointed to mistakes and difficulties the students had. Given categories of behavior can be applied to development of questionnaire intended for observation of problem-solving process in this context. Identified mistakes, difficulties and students' behavior patterns can contribute at great extent to the understanding of students' way of thinking, and therefore contribute to the way of organizing teaching and planning content in this field.*

Keywords: *programming; code.org; problem solving; categories of behavior during solving problems*

1. INTRODUCTION

The development of algorithmic thinking has become a frequent topic when it comes to elementary education. It is known that learning programming languages can be complex for students of all ages and particularly for younger students who have not yet developed thinking skills. Therefore, encouraging logical thinking is important at younger age. Programming skills are the integral part of digital literacy and computing thinking [1]. Nevertheless, these skills should not be observed only in the context of digital literacy but as transversal skill applicable to other spheres of life and studying. The researches imply that learning programming at early age has positive effect on mathematical, logical [2], critical thinking [3], computing thinking, problem solving skills, creativity and imagination [4].

Syntax writing which significantly complexify the essential development of algorithmic way of thinking and solving problems is replaced with the usage of graphical environment for learning programming. Such learning environments enable students to focus on the very problem and its solving instead on syntax writing. Graphical environments are based on writing programs by building blocks using swiping technique. There is a great number of tools for visual learning of programming such as Scratch, Alice, MIT App Inventor and others. Code.org platform will be described for the purpose of this work.

Code.org is a tool for learning programming organized through courses adjusted to preschool and school children. Unlike other platforms this tool consists of beforehand structured tasks of different complexity distributed through tasks paths. Each task path includes one programming concepts such as sequences, loops, conditions and similar. The paths consist of tasks of different complexity and solving more complex tasks does not depend on prior solving of simpler ones. The environment supports monitoring of program correctness by its performing which is shown graphically. It is possible to test the program in its entirety or "step by step "where every block is marked in the program currently done. Thus, the students are enabled to find mistakes in the code. Certain tasks which are included in the courses require finding incorrect blocks in already offered programs.

Programming consists of numerous thinking activities and therefore it is necessary to determine the way in which thinking process in younger students while solving problems is carried out. Younger students (in the first cycle of elementary education) can encounter difficulties in abstract thinking and hence be limited to solving concrete, practical problems [5].

Therefore, it is important to determine difficulties students encounter when solving problems, mistakes they make and other additional behaviors which can explain problematic points in the process of solving problems. Clear insight into learning process can, at great extent contribute to better organization of teaching and content adjustment in this field. The overview of recent research implies that there are no studies which deal with examination of the solving problem process within the context of programming of 4^{th} grade elementary students.

Therefore, the aim of this work is to identify categories of expected behavior during solving problems of fourth grade students and development a questionnaire for monitoring process of solving problems.

2. SOLVING PROBLEMS WITHIN THE CONTEXT OF PROGRAMMING

Analog thinking can have great impact on solving problem process [6]. Mental elements, as basic units of thinking process, and combinations of the same represent the key component in the process of solving problems [7], and very determination of relations between the elements is an essential part of analog thinking. Analog thinking is explained as a process of representation of objects and information as a system of mutual connections which can be compared and combined [8]. Graphical environments for learning programming are based on building blocks, which are elements necessary for correct execution of the program where it is essential to connect relations of more blocks. In addition, when a student solve tasks which are linked by content or when more complex tasks require necessary understanding of simpler ones then analog thinking becomes more prominent. Analog thinking is recognized by its usage of existing information from long-term memory, reorganization, modification and its usage in new situations. Therefore, analog thinking consists of the following phases [9]: redirecting attention to important information, determining relations between elements, domain or elements mapping, bringing conclusions and establishing mutual principles between comparable domains. Beside analog thinking, there are following strategies for solving problems [10]: one-trial solving, multiple trial problem solving, solving problems by complete application of information leading to correct solution, partial application of information leading to incorrect solution, solving problems by testing and making mistakes and systematic testing. From the above mentioned it can be concluded that the mentioned strategies can be roughly categorized into efficient and less efficient solving problems strategies. One-trial solving can be followed by systematic testing, that is planned and targeted problem solving. Solving problems by testing and random tries is less efficient strategy which can lead to solving problems by multiple trying and mistakes repetition. While monitoring the way of solving problems using computer (simulation) the following behaviour categories are seen [11]: understanding the aim of the task before attempting solving the same; tendency to making mistakes and random tries; steps repetition; perseverance. Research has shown that during solving problems by using mind game, students age 7-11 have a tendency to solve problems by random tries, making mistakes, without prior planning [12]. Other studies have shown that these strategies for solving mathematical problems are applied by students of lower level of knowledge [13].

In relation to the above mentioned there are metacognitive processes which are integral part of the problem solving process and consist of the following subprocesses [7]: establishing the very nature of a problem; determining steps or necessary components during solving problems; determining strategies for establishing order of elements while solving problems; the choice of information mental structure; resource distribution; solution monitoring.

Other regulation strategies can be also active during problem solving and they are often, in children, manifested in using private speech [14]. According to some opinions, if the problem tasks are too difficult, the students can lose control over regulation process. Therefore, it is necessary to set optimal complexity of tasks which will consequently encourage regulatory processes in students [15]. In addition, one more strategy of self-regulation is seeking for help. However, some students do not seek for help and they do not ask additional questions even though they have difficulties when doing tasks [16].

The mistakes students make when solving problems within the context of programming can at great extent contribute to understanding of their way of thinking, as well as to determining problematic programming concepts. Recent research has shown that students encounter difficulties in learning specific programming concepts such as loops, nested loops, branching etc. [4]. The results have shown that students of 5^{th} and 6^{th} grade understand simple form of loops whereas they have difficulties in understanding nested loops [17]. Mladenovic and others [18], also find out that students reason incorrectly when it is expected to determine the number of repetitions of nested loops. As expected, the results of studies point to the fact that students firstly adopt sequencing problem solving [4], [19]. The analysis of students' projects in Scratch (age 8-11), refers to the fact that almost all students' projects contained sequencing approach, and more than half projects contained iterations [19].

Difficulties in solving problem tasks and analysis of follow-up behaviors can provide clearer picture of presence of solving problem patterns within the context of programming in students in the first cycle of elementary education. Therefore, it is important to develop valid instrument for evaluation of these behaviors.

3. RESEARCH METHODOLOGY

The main objective of this research is to determine the 4th grade students' difficulties during solving problem tasks within the context of programming which require application of certain strategies, analogies, algorithmic thinking, perseverance, etc. Therefore, the integral part of the research is forming behavior categories which can be used in a questionnaire for monitoring solving problem process.

Research variables are expected mistakes and behaviors for three tasks groups:

- Mistakes students made in all three tasks groups (nine types of mistakes in total);
- Independence in finding analogies and mistakes;
- Making mistakes timing (in initial tasks, later tasks and always);
- Variety of mistakes (same, different, same and different);
- Solving tasks by random guessing;
- Number of tries of solving tasks by random guessing;
- Change of steps when solving problems, understanding of change of steps;
- Evaluation by commenting and independence in it;
- Demonstration of planned steps, independence in it and accuracy;
- Determining the number of loop repetition, number of "repeat" blocks;
- Determining necessary number of "repeat" blocks;
- Locating "repeat" blocks, use of combined, sequencing solution and solving tasks by application of nested loops;
- Understanding the relation between blocks, looking for new solution and independence in it;
- Number of tries of new solution application;
- Approaching new solution without demand.

3.1. Methods and techniques for collecting data

The selection of tasks which students will solve in code.org environments is done in the first phase. The tasks are chosen in the way that they comprise several programming concepts. Course E, which is intended for that age, is used for 4th grade students sample (https://studio.code.org/s/coursee-2018). The tasks students solved were divided in three tasks groups depending on their complexity and programming concept. The first group of tasks consisted of concept of sequencing programming. Within this group the students solved five tasks. The students were required to add blocks in already existing program in order to perform task correctly and find mistakes in offered solutions. In the other group of tasks, the students were required to draw geometric shapes by using blocks. The demands referred to accurate determining of object movement direction, accurate blocks arrangement and identification of other possible solution by using loops in order to reduce the number of blocks.

Within this group, the students solved four tasks. The third group of tasks meant solving problems by nested loops. The tasks required determining number of loops repetition, places of using nested loops and accurate arrangement of blocks realized outside the loops. The students did two tasks in this group of tasks.

Then a research in which students individually went through all categories of tasks with researcher was conducted. The procedure of solving problems and conversation between a researcher and students were video recorded. In order to ensure that all students can do the same number of tasks in determined time a pilot study with two students of different school achievement was conducted. The students first got familiar with environment and the way of using platform. The researcher gave students tasks in printed form so that they could have insight into task requirements in Serbian language. Before very start of solving tasks, the students read tasks aloud. At the beginning the students were explained that they were supposed to do the tasks individually, but they could also ask for help or additional explanations. The students were explained that they were supposed to talk aloud while solving problems. When needed, the researcher encouraged students to think and redirected their attention to the task. Work with every student lasted for approximately one hour. In the final phase, based upon the transcripts which were recorded, students' mistakes and behaviors were recognized.

The sample consisted of 29 students of fourth grade from two elementary schools: „Milan Blagojević " in Lučani and „Tanasko Rajić" from Čačak. 14 girls and 15 boys of different school achievement took part in the research.

4. RESULTS AND DISCUSSION

This chapter gives overview of research results followed by discussion. The first chapter is about qualitative analysis of transcript and the second chapter one is about separate behavior categories based on qualitative analysis.

4.1. Qualitative analysis of transcript

Qualitative analysis of video materials, that is transcript, refers to manifestation of mistakes students made, as well as follow-up behaviors occurring during the process of problem solving. An example of transcript which describes the process of solving tasks by applying sequencing solution and loops is in the following text:

An example of solving tasks in in the second path where the student is required to apply both sequencing and loop solution:

A student is reading task text: An artist has to walk over shaded path like in previous task but changing the color of path all five times.

After a break the researcher is checking understanding of the task objective: „*What are we supposed to do in the task?* "

*The student reads silently the task text. *

*The student starts to solve the task by sequencing solution (that is, he doesn't recognize other possible solution by himself). *

The student leaves out the block „change the color ". (the students fail to understand task objective)

The researcher reminds the student of task objective: „*Shifting each step the color of field should be changed*. "

* The student is reading silently the task text again. *

The students ask a question: „*Am I supposed to change the color here*? "(*an evaluation of his own way of thinking is done*)

*The researcher confirms. *

The student comments: „*Then, it means that I drag the block here „set color* … "(step verbalization)

„…*Then moves forward, then he should change the color again …* "(the student repeats it three times), „*I think that is it!* "(the student performs evaluation)

The researcher reminds student that he can try the problem solution: „*Would you like to try solution?* "

*The student tries to execute the program by clicking „Run"and understands his mistake. * He comments: „*Ah, it means that I … "and adds some more necessary blocks* (planned steps are accurate)

*The student now independently tries out the tasks and concludes that it is necessary to add two more blocks. *

*The student manages to solve the task. *

*The researcher now demands that student should apply other solution, that is perform the task by using fewer blocks. The researcher suggests application of new block. *

Since he realizes there is no feedback information, the researcher encourages student to think: „*Have you noticed that some blocks are repeating*? "

The student answers: „*Yes, move forward and set color* ".

The researcher: „*What do you think, which block we haven't used so far, and we can use in order to finish the task by using fewer blocks?* "

The student: „*Well, maybe „jump forward* ", „*No, move to the right* "… "(the student doesn't understand the task, randomly chooses correct answer)

The researcher helps the student: „ *Can we use this block "repeat"* '?

The student: „*Ah, repeat four times.* "*Choose a block and drags under all other blocks. *

*The researcher repeats the task objective. *

The student: „Now, possibly the block „*set pattern*. "(misunderstanding continues)

The researcher repeats task objective and sets the question: „*How many times should blocks be repeated, move forward and change color*? "

The student: „*9 times.* "(the student counts the blocks)

The researcher: „*That is a total number of blocks, but how many times have blocks, move forward and change color repeated?* "

The student: „*5 times* "(the student gives correct answer)

The researcher continues to point to possible solution: „*Then, what should a shrimp do 5 times?* "

The student: „*Ahem, five times …* "„*Five times to set color, which means that him (object) five times* "(even though the student understands at the beginning which blocks are repeated several times, now he cannot determine the same)

*The researcher offers help by removing additional blocks, placing loop in its correct position and writes down the correct number of loop repetition. The student is now asked to continue with task solving. *

The student comments: „*Move forward is repeated five times* … "

The researcher encourages the student to solve the task: „*Come on, show what you have just said*. "

*The student continues to build blocks under the block repeat one beneath the other and apply sequencing solution (Figure 1). *

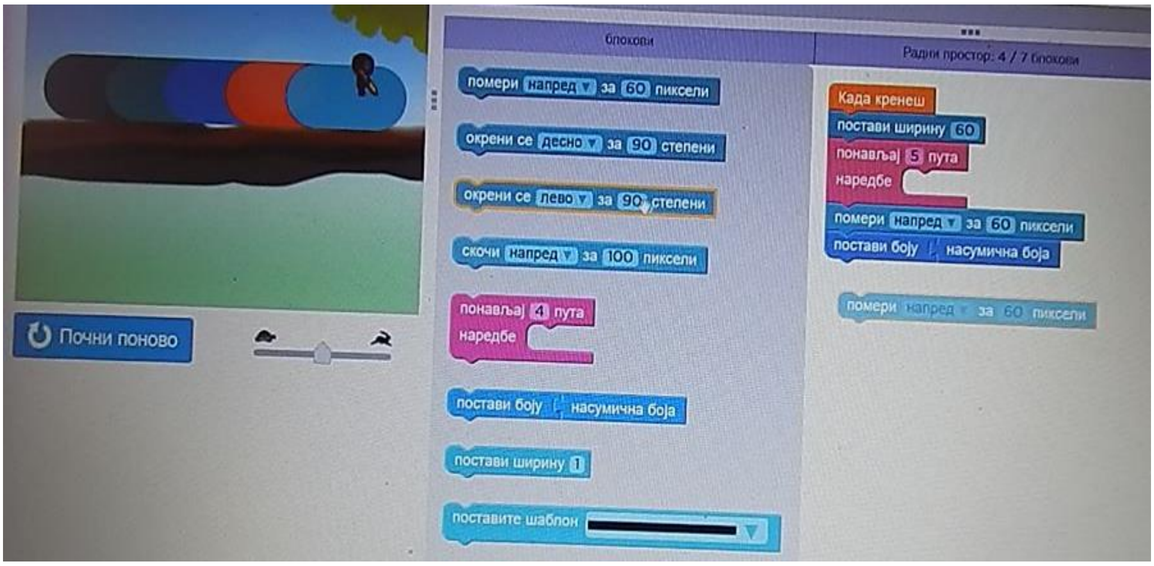

Figure 1. *Mistakes in arrangement of blocks (inclination for applying sequencing solution)*

Certain behaviors and students' way of thinking while solving problems can be seen in above mentioned example. Certain behaviors which can be recognized in this example (and which are included in the questionnaire) refer to misunderstanding of task objective, oral evaluation of accuracy, oral verbalization of steps which are accurate and partially accurate, random guessing of correct solution, difficulties in identifying other possible solution and general understanding of task demands. Recognized mistakes in this example made by the student are the following: inclination for applying sequencing solution and besides added repeat block there is a mistake in identifying which blocks are executed in the loop, that is outside the loop , a mistake in loop positioning and how many times the loop is repeated.

In relation to the mistakes the student made misunderstanding of very task can be confirmed and therefore during the application of sequencing solution the student needed initial leading by the researcher. Trying out solution accuracy, making conclusions and applying planned steps the student finally managed to complete the task independently by using sequencing. This student also did an evaluation of his way of thinking by asking the researcher questions. However, when the student was asked to identify other possible solutions, due to task misunderstanding the student randomly chose (incorrect) solutions. Therefore, in the second phase of solving problem, due to more complex demand, the student had a tendency to find solution by random tries. After researcher's suggestion to choose "repeat" block, the student accepted mechanically the suggestion without understanding. It can be seen in the student's attempt to use "repeat" block where the number of repetitions is four. Besides, the student kept all previously added blocks outside loop which refers to wrong reasoning when it was supposed to decide how many times to use two necessary blocks. Even though the student managed to determine the number of calls of these blocks during application of sequencing solution, during application of loop he was not able to apply independently analogy to determining blocks which are executed five times.

When the researcher removed extra blocks (outside loop) and encouraged student to think over, the student wrote the correct number of repetitions, but still by applying sequencing solution under "repeat" block (Fig. 1). Besides making mistakes in identifying which blocks are executed inside the loop, the student showed additional misunderstanding by continuing placing two blocks five times each like in sequencing solution. These mistakes have shown misunderstanding of relations between blocks.

This finding is in accordance with other authors' findings which imply that students adopt first the concept of sequencing programming whereas more complex concepts are more difficult for understanding in younger students [4]. Other research also implies that students (age 7-11) when solving problems which require more thinking are inclined to adopt unplanned approach of solving problems, repeat mistakes and apply strategy of random trying [12]. On the other hand, the respondents did not have any prior knowledge in the field of programming which could also influence application of less efficient problem-solving strategies [13].

4.2. Students' behavior categories during solving problems in the context of programming – development of questionnaire

The analysis of transcript refers to students' behaviors during solving problems and those behaviors can be categorized. Most categories can be mutual to all three groups of tasks whereas certain ones need to be adjusted to specific programming concept. Based upon shown and other transcripts the following behavior categories can be seen:

- Trying out and changing strategies category for checking frequency of mistakes repetition, type of mistakes, time when the mistakes occur (at the beginning, during doing more complex tasks , always); frequency of solving tasks by random guessing and number of tries; presence of demonstration of planned steps and independence, their accuracy; presence of changing steps during solving problems and understanding of the same; presence of skills needed for usage of analogy and independence in it;
- Locating mistake (only for the first group of tasks) category for testing the capability of students to find a mistake in offered program which does not function correctly;
- Code tracking and evaluation category for checking understanding relations between blocks; presence of oral evaluation and independence in it;
- Repetition of objective is a category for checking students' tendency to interpret wrongly task setting and independence of evaluation of the solution by asking questions;
- Showing uncertainty in taking steps category for determining whether students show uncertainty while solving tasks in situations when they encounter difficulties;
- Looking for other solution (in the second and third group of tasks) category for evaluating the students' skills for finding other solutions and their independence in it, as well as correct application of other solution;
- Understanding loops (in the second group of tasks) is a category which was used for checking independence in determining number of commands repetition, places where it is necessary to insert a loop , mistakes that can occur (wrong adding of commands executed outside "repeat" block ; beside usage of "repeat" block the student uses the same set of commands several times; mistakes in identifying which blocks are executed outside loop) and inclination to applying sequencing solution;
- Understanding nested loops (only in third group of tasks) is a category which was used for checking independence in determining number of necessary "repeat" blocks, locating these blocks, determining the number of repetitions; inclination and independence in applying sequencing, combined solutions or nested loops.

In order for further monitoring problem solving process and checking validity and reliability of observed behaviors, these mentioned categories were included in questionnaire. An example of one item for questionnaire is given in the following:

> An example of questionnaire item:
>
> <u>Tracking code and evaluation (Student follows task parts and analyses blocks: „We have added one step move forward: it means we need one more. "; he checks his solution)</u>
>
> The student shows understanding of relations between blocks (the importance of order of blocks – e.g. accuracy in answering question „Where should we insert repeating commands when "repeat" block is used? "(before, after, inside the block)?).
>
> *While determining whether student understands relation between blocks, the student:* (a) does not show understanding at all (b) shows understanding with researcher's help (c) completely understands order and purpose of certain blocks

5. CONCLUSION

In accordance to shown, it can be concluded that process of solving problems in the field of programming in younger students is important to be understood better in order to adjust these contents to their age. Mistakes made by students show how student thinks. Therefore, identification of mistakes is vital for finding out problematic points in understanding students.

Given transcript example indicates behavior categories which are observed in the thinking process of solving programming tasks in students age 10 and which can be used in a questionnaire for monitoring process of problem solving in this context. The categories refer to strategies of solving problems, understanding programming concepts, skills for finding other solutions, presenting uncertainty, tracking code and evaluation, task understanding, as well as mistakes students made.

Nevertheless, it is necessary to conduct questionnaire validity check in relation to assessments by several observers, and then by statistical analyses. Usage of questionnaire would ensure quantitative data analysis and as a consequence a clearer picture of problem-solving process would be provided. Moreover, it would be useful to monitor the same behavior categories while using other platforms for learning programming.

ACKNOWLEDGEMENTS

This study was supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia, and these results are parts of the Grant No. 451-03-68/2020-14/200132 with University of Kragujevac - Faculty of Technical Sciences Čačak.

REFERENCES

[1] Papadakis, S., Kalogiannakis, M., & Zaranis, N. (2016). Developing fundamental programming concepts and computational thinking with ScratchJr in preschool education: a case study. *International Journal of Mobile Learning and Organisation*, 10(3), 187-202.

[2] Strawhacker, A., Lee, M., Caine, C., & Bers, M. (2015, June). ScratchJr Demo: A coding language for Kindergarten. In *Proceedings of the 14th International Conference on Interaction Design and Children* (pp. 414-417). ACM.

[3] Su, A. Y., Yang, S. J., Hwang, W. Y., Huang, C. S., & Tern, M. Y. (2014). Investigating the role of computer-supported annotation in problem-solving-based teaching: An empirical study of a Scratch programming pedagogy. *British Journal of Educational Technology*, 45(4), 647-665.

[4] Fokides, E., & Atsikpasi, P. (2017). Redefining the framework for teaching programming to primary school students: Results from three pilot projects. *British Journal of Education, Society & Behavioural Science*, 20, 1-11.

[5] Zaharija, G., Mladenović, S., & Boljat, I. (2013). Introducing basic programming concepts to elementary school children. *Procedia-social and behavioral sciences*, 106, 1576-1584.

[6] Clement, C. A., Kurland, D. M., Mawby, R., & Pea, R. D. (1986). Analogical reasoning and computer programming. Journal of *Educational Computing Research*, 2(4), 473-486.

[7] Antonietti, A., Ignazi, S., & Perego, P. (2000). Metacognitive knowledge about problem-solving methods. *British Journal of Educational Psychology*, 70(1), 1-16.

[8] Richland, L. E., & Simms, N. (2015). Analogy, higher order thinking, and education. Wiley Interdisciplinary *Reviews: Cognitive Science*, 6(2), 177-192.

[9] Vendetti, M. S., Matlen, B. J., Richland, L. E., & Bunge, S. A. (2015). Analogical reasoning in the classroom: Insights from cognitive science. *Mind, Brain, and Education*, 9(2), 100-106.

[10] Kolovou, A., & Heuvel-Panhuizen, M. V. D. (2010). Online game-generated feedback as a way to support early algebraic reasoning. *International Journal of Continuing Engineering Education and Life-Long Learning*, 20(2), 224-238.

[11] Zoanetti, N. (2010). Interactive computer based assessment tasks: How problem-solving process data can inform instruction. *Australasian Journal of Educational Technology*, 26(5), 585-606.

[12] Bottino, R. M., & Ott, M. (2006). Mind games, reasoning skills, and the primary school curriculum. *Learning Media and Technology*, 31(4), 359-375.

[13] Elia, I., van den Heuvel-Panhuizen, M., & Kolovou, A. (2009). Exploring strategy use and strategy flexibility in non-routine problem solving by primary school high achievers in mathematics. *ZDM*, 41(5), 605-618.

[14] Vygotsky, L. (1978). Interaction between learning and development. *Readings on the development of children*, 23(3), 34-41.

[15] Behrend, D. A., Rosengren, K., & Perlmutter, M. (1989). A new look at children's private speech: The effects of age, task difficulty, and parent presence. *International Journal of Behavioral Development*, 12(3), 305-320.

[16] Schunk, D. (Ed.), Zimmerman, B. (Ed.). (2008). *Motivation and Self-Regulated Learning*. New York: Routledge, https://doi.org/10.4324/9780203831076

[17] Giannakoulas, A., & Xinogalos, S. (2018). A pilot study on the effectiveness and acceptance of an educational game for teaching programming concepts to primary school students. *Education and Information Technologies*, 23(5), 2029-2052.

[18] Mladenović, M., Rosić, M., & Mladenović, S. (2016). Comparing elementary students' programming success based on programming environment. *International journal of modern education and computer science*, 8(8), 1-10

[19] Wilson, A., Hainey, T., & Connolly, T. M. (2013). Using Scratch with primary school children: an evaluation of games constructed to gauge understanding of programming concepts. *International Journal of Game-Based Learning (IJGBL)*, 3(1), 93-109.