Journal of Teacher Education and Lifelong Learning (TELL)

Volume: 5 Issue:1 Year: 2023 | Research Article | ISSN: 2687-5713

Examination of Projects Prepared by Prospective Secondary School Mathematics Teachers Using Scratch

Selva Büşra TURAN[1] Ahmet ERDOĞAN[2]

[1]Necmettin Erbakan University, Ahmet Kelesoglu Educational Faculty, Department of Mathematics and Science Education, Konya, Türkiye
sbturan@erbakan.edu.tr

[2]Necmettin Erbakan University, Ahmet Kelesoglu Educational Faculty, Department of Mathematics and Science Education, Konya, Türkiye
aerdogan@erbakan.edu.tr

Article Info

Article History
Received: 13/12/2022
Accepted: 20/04/2023
Published: 30/06/2023

Keywords:
Prospective Mathematics Teachers,
Scratch,
Programming Concepts,
Computational Thinking Concepts.

ABSTRACT

The aim of this study is to examine the projects prepared by prospective secondary school mathematics teachers using Scratch for the learning outcomes in the Mathematics Course Curriculum (Secondary School 5th, 6th, 7th, and 8th grades) in terms of the proficiency level of programming concepts and computational thinking concepts. 73 prospective secondary school mathematics teachers participated in this research in which the case study design was used. Prospective teachers prepared Scratch projects related to the 5th, 6th, 7th and 8th-grade level learning outcomes of the Mathematics Course Curriculum. A total of 292 Scratch projects were examined within the scope of the research. Scratch projects were evaluated through the "Scratch Projects Assessment Rubric" and the "Dr. Scratch Assessment Tool". The result of the research showed that most of the criteria in the "Scratch Projects Assessment Rubric" were provided in more projects than at the beginning as new projects were developed by the prospective teachers at different grade levels every week. Nevertheless, it was evident from the Dr. Scratch evaluation that as the prospective instructors utilized the Scratch program, the level of proficiency of the projects they created is improved. It is recognized that this beneficial development has occurred because of the rise in the number of potential instructors who have experience with the Scratch program.

Citation: Turan, S.B. & Erdoğan, A. (2023). Examination of projects prepared by prospective secondary school mathematics teachers using scratch. *Journal of Teacher Education and Lifelong Learning*, *5*(1), 209-221.

[1]This study was produced from the first author's doctoral dissertation. In addition, this study was presented a paper in the International Education Congress (EDUCongress 2022) on 17-19 November 2022 in Antalya.

INTRODUCTION

In today's world where information technology is developing very rapidly, the computer shows its effect as an indispensable tool in all areas of life increasingly. Programming, which is an indispensable component of computers, is also progressing by diversifying in parallel with this visible and developing effect of computers (Erümit & Berigel, 2018). In line with these developments, a wide range of computer programs that will be beneficial for the students is developed and used in education, such as ensuring the active participation of the students in the course, keeping their interest in the course alive, and increasing their motivation (Öztürk, 2021). Integration of technology into education is to make technology an indispensable element of education processes in accordance with learning objectives as well as incorporating technology into education (Atun & Usta, 2019).

Overcoming rapid changes and making sense of the world requires not only understanding how technology is advancing but also acquiring and developing skills that will help adapt to these changes (Erdoğan & Şimşek, 2018). In this sense, programming is accepted as one of the 21st century skills that everyone should have (Yükseltürk & Üçgül, 2018). Programming refers to all activities from the design of a solution to its implementation in the process of solving a problem (Karal et al., 2018). Another concept that is not considered separately from this concept and affects each other is the computational thinking skill when it comes to programming (Üzümcü & Bay, 2018). Computational Thinking refers to problem-solving, designing systems, and understanding human behavior by making use of the basic concepts of computer science. Computational thinking is a fundamental skill that should be known not only by computer scientists but also by different segments of the population concerned (Wing, 2006).

In the world evolving into the information age, new technologies that are inevitably bound to be widely used, lead to updates in the education-teaching process by providing new opportunities (Ersoy, 1997; Oluk & Çakır, 2021). While these new tools enable to restructure of the processes of learning and teaching mathematics, they also guide the expectations from mathematics and the way they use mathematics (MONE, 2018). In the literature, it is seen that the contribution of technological tools to mathematics education is widely included: Software that draws graphs, computer algebra systems, software with a programming language, and graphing calculators are some of them. In addition, it is becoming easier for students to visualize mathematical concepts and to have access to multiple representations of concepts quickly and effectively (Akkoç, 2013). In recent years, many countries have realized the importance of programming and started to teach programming either as an independent course or by integrating it into different subjects such as mathematics and physics (Apriola & Tedre, 2012, as cited in Şimşek, 2018). In particular, there is a remarkable relationship between mathematics and programming. In the context of this relationship, it is possible to associate the programming curriculum with the mathematics course outcomes and enrich the content of the courses (Lewis & Shah, 2012). In addition, it is possible to teach some abstract and difficult-to-comprehend content in mathematics programs more easily through programming (Akpınar & Altun, 2014).

Many students struggle with learning programming (Gomez & Mendes, 2007). Traditional programming languages such as C, C++ are difficult for students to learn (Genç & Karakuş, 2011). Traditional programming languages are structurally difficult, and complex, can cause some difficulties. Recently, open-source platforms such as Scratch, Google Blockly, and Code.org, which make programming easier, user-friendly, and supported by many visual features, have been developed to guide and encourage candidates who are just starting to learn programming (Aytekin et al., 2018). Among these platforms, Scratch is a block-based visual programming environment that allows the creation of interactive and communication-rich projects. In fact, Scratch was initially used in informal learning environments, then it is increasingly used as an educational teaching material in schools (Maloney et al., 2010). The fact that Scratch is suitable for easy and fast learning, as well as being easy-accessible and allowing those who want to learn to improve their programming skills, increases the frequency of its application in the field of education (Iskrenovic-Momcilovic, 2020). The Scratch program provides students with an environment where they can enjoy programming and exploration and creativity, while also supporting them to improve their understanding of embedded programming and mathematical concepts (Calder, 2018). Scratch can be used as a tool to support students' learning while simultaneously facilitating teachers' teaching efforts (Mo et al., 2021).

The development of these easier-to-use programming languages provides educators with significant opportunities how to use these programs to teach math more easily and with fun (Germia & Panorkou, 2020). In this sense, while discussing the importance of programming in the educational process, it should not be ignored that teachers and prospective teachers are at the center of this process (Güleryüz et al., 2020); because one of the most important functions of education is to raise individuals equipped with the skills required by their era (Doğan, 2014). The development of prospective teachers skills in using technology can be considered a critical variable that improves the quality of the education system (Usta & Korkmaz, 2010). In this sense, it is important and necessary to train prospective teachers who are equipped with the skills required by the information age due to their critical roles in the education process and the mission they have (Güleryüz et al., 2020). In the literature, there are studies evaluating Scratch projects. It is seen that Scratch projects are evaluated in terms of pedagogical aspects (Öztürk, 2021), in terms of design elements and educational aspects (Yıldız Durak and Karaoğlan Yılmaz, 2019), in terms of programming concepts (Yıldız Durak et al.,2018; Gabriele et al., 2019; Öztürk, 2021) or in terms of computational thinking concepts (Gabriele et al., 2019) in these studies. Considering the studies that evaluate the projects in terms of programming concepts or proficiency level of computational thinking concepts; it is seen that study groups include primary teacher education prospective teachers (Gabriele et al., 2019), students of psychological counseling and guidance and social studies teaching department (Yıldız Durak et al., 2018), mathematics teachers (Öztürk, 2021), prospective kindergarten teachers (Papadakis and Kalogiannakis). In this sense, it is understood that the studies conducted with prospective secondary school mathematics teachers are incomplete. Due to its explained importance and determined deficiency, in this study, it is aimed to examine the projects prepared by prospective secondary school mathematics teachers, using Scratch for the learning outcomes in the Mathematics Course Curriculum (Secondary School 5^{th}, 6^{th}, 7^{th}, and 8^{th} grades) in terms of programming concepts and competency level of computational thinking concepts. In line with this purpose, the following questions were tried to be answered:

1. Which programming concepts do prospective secondary school mathematics teachers include in the projects they prepare by using Scratch for the learning outcomes in the Mathematics Course Curriculum (Secondary School 5^{th}, 6^{th}, 7^{th}, and 8^{th} Grades)?

2. What is the Proficiency Level of Computational Thinking Concepts in the projects prepared by prospective secondary school mathematics teachers using Scratch for the learning outcomes in the Mathematics Course Curriculum (Secondary School 5^{th}, 6^{th}, 7^{th}, and 8^{th} Grades)?

METHOD

Research Design

Case study design was used in the study. The purpose of case studies is to reveal the results related to a particular situation. In this study, the holistic single case design, one of the case study designs, was used. In holistic single case studies, there is only one analysis unit, and it is aimed to interpret a situation holistically in its natural environment (Yıldırım & Şimşek, 2006).

Research Study Group

73 prospective teachers who took the Algorithm and Programming course in the 2nd grade of the secondary school mathematics teacher education undergraduate program of a state university participated in the study. Since the pre-service teachers took Algorithm and Programming course in the 2nd grade, the study was carried out with 2nd grade students. Purposive sampling, one of the non-probability-based sampling methods, was used in the study.

Table 1. *Distribution of the study group by gender*

GENDER	f	%
Female	55	75.3
Male	18	24.7
Total	73	100

Examining Table 1, it can be seen that 75.3% of the prospective teachers participating in the study are female and 24.7% are male students.

Research Instruments and Processes

The study was carried out within the scope of the "Algorithm and Programming" course taken by prospective secondary school mathematics teachers for one semester. Due to the COVID-19 outbreak, the study was carried out synchronously through the distance education system of the university. Moreover, in addition to the live sessions, video footage was made by the researcher in which the Scratch program was narrated, and those videos were uploaded to the researcher's personal YouTube channel. The links to the uploaded videos were sent to the prospective teachers through the distance education system. The deficiencies of the Scratch program were eliminated during the live class hours by asking the researcher about the issues that the prospective teachers did not understand in the video recordings they watched. Prospective teachers started to prepare Scratch projects, starting from the 5th-grade level, at the 6th, 7th, and 8th-grade levels respectively, in accordance with the learning outcomes of the 2018 Mathematics Course Curriculum (Secondary School 5th, 6th, 7th and 8th Grades), after the completion of the Scratch program courses. The projects at the 5th, 6th, 7th, and 8th-grade levels, which were prepared by each prospective teacher, were uploaded on both the distance education system and to the Scratch studios opened by the researcher on the Scratch website. The prospective teachers uploaded Scratch projects to the created studios, which were selected based on various learning outcomes. This allowed them to observe, assess, and provide feedback on projects that were designed with different learning objectives. Moreover, they were able to generate new ideas and concepts for the projects they would create for the subsequent grade level. Each of the 73 pre-service teachers prepared 4 Scratch projects, one for each grade level (5th, 6th, 7th, and 8th grade). Thus a total of 292 Scratch projects were prepared by prospective teachers at the end of the study. The projects prepared by the prospective teachers regarding the learning outcomes in the Mathematics Course Curriculum (Secondary School 5th, 6th, 7th, and 8th Grades) were analyzed with the "Scratch Projects Assessment Rubric" and "Dr. Scratch Assessment Tool".

Scratch Projects Assessment Rubric

A coding schema has been developed by Denner et al. (2012) that is thought to correspond to computer science programming concepts and that will be used to determine the extent to which these concepts are used in the content when content is created. The coding scheme consists of 3 main categories and 24 subcategories. Gabriele et al. (2019) adapted this coding scheme according to Scratch and organized it into 3 main and 17 subcategories. The first main category, "Programming Concepts", consists of 9 subcategories, the second main category, "Code Organization", 3 subcategories, and the third main category, "Designing for Usability", consists of 5 subcategories. In the developed project, 1 or 0 points are given according to whether each item is present or not, and the project is evaluated. This assessment rubric also reveals which computational thinking concepts are learned by users (Gabriele et al., 2019). Denner et al. (2012) refer to the categories of "Programming Concepts", "Code Organization" and "Designing for Usability" in this coding scheme as 3 competencies that they determine to engage individuals with computational thinking, and these competencies are evaluated with this coding scheme.

Dr. Scratch Assessment Tool

Dr. Scratch is a free open-source web application that allows you to easily analyze Scratch projects as well as obtain feedback that can be used to improve programming skills and computational thinking. To analyze a project with Dr. Scratch, simply copying the project's Uniform Resource Loader (URL) is sufficient. Analyzing a Scratch project using Dr. Scratch, it shows the user the computational thinking score. Dr. Scratch assesses proficiency in seven concepts. These are flow control, data presentation, abstraction and parsing, user interaction, synchronization, parallelism, and logic. A project is evaluated on a scale of 0-21 and each competency is evaluated on a scale of 0-3 (Moreno Leon et al., 2015). Projects are accepted at the "Basic" level between 0-7 points, "Developing" level between 8-14 points, and "Proficiency" level between 15-21 points according to the score they receive (Moreno Leon & Robles, 2015). The application also reveals situations such as bad programming habits, code repetitions, and codes that never work (Demir & Seferoğlu, 2017). Dr. Scratch also provides users with a link to download project certificates that show the score they received from their project (Setyawan, 2020).

Data Analysis

The projects prepared by the prospective teachers according to the 5th, 6th, 7th, and 8th-grade curriculum learning outcomes were analyzed with Dr. Scratch Assessment Tool and Scratch Projects Assessment Rubric, and descriptive statistics (percentage, frequency, average) were included. The evaluation of the projects was conducted using the Scratch Projects Evaluation Rubric, with the researcher, a field expert, and a computer engineer expert analyzing the projects. To ensure the reliability of the data analysis, a formula [Consensus / (Consensus + Disagreement) x 100] was utilized (Miles & Huberman, 1994). As a result of this independent verification, 96% inter-rater reliability was achieved. In cases where there are different encodings, the coders are united on a common opinion and codified. Sample Project Evaluation with Scratch Projects Assessment Rubric is given below.

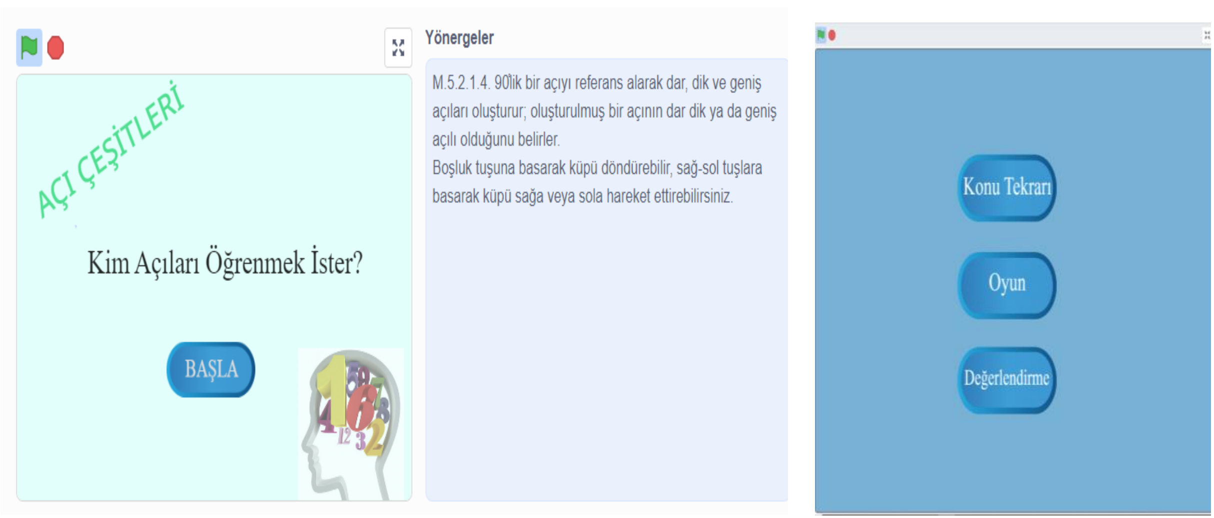

Figure 1. *Example of a scratch project prepared by a prospective teacher*

The project, which was prepared by choosing the learning outcome of M.5.2.1.4 from the 5th grade geometry and measurement learning field, was prepared in three parts as subject repetition, game, and evaluation. It is seen that the sequence, user interaction, iteration/loop, variables, conditional states, coordination and synchronization, random numbers and boolean logic criteria are included in the

Programming Concepts main category when the project is examined according to the Scratch projects assessment rubric. Looking at the Code Organization main category, it is seen that there is no extraneous block used in the project. In the game part of the project, a variable was created and named "*score*" in accordance with its purpose, but the names of most sprite used in the project were left as they were in the program's library. Assessing the criteria within the Design for Usability main category, it is seen that the program has been meticulously developed to align with its intended purpose, as determined by the selected learning outcome. The program operates seamlessly, meeting the functionality criterion. Furthermore, the project encompasses user customization of sprites and stage, with a clear and concise explanation of its operations, and it stands out as an original creation. Accordingly, the scoring of the project is given Table 2.

Table 2. *Scoring of the project*

PROGRAMMING CONCEPTS	***f***
1. Sequence	1
2. User interaction	1
3. Iteration / Loop	1
4. Variables	1
5. Conditional statements	1
6. Lists (arrays)	0
7. Coordination and synchronization	1
8. Random numbers	1
9. Boolean logic	1
CODE ORGANIZATION	
10. Extraneous Blocks	0
11. Sprite names	0
12. Variable names	1
DESIGNING FOR USABILITY	
13. Functionality	1
14. Sprite customization	1
15. Stage customization	1
16. Clear instructions	1
17. App originality	1

Ethic

For the research, the ethics committee approval was obtained from Necmettin Erbakan University, Social Sciences, and Humanities Scientific Research Ethics Committee with the decision dated 19.02.2021 and numbered 2021/83.

RESULTS

The Scratch Projects Assessment Rubric adapted by Gabriele et al. (2019) was used to analyze the programming concepts used by prospective teachers in their Scratch projects related to, respectively, 5th, 6th, 7th, and 8th-grade learning outcomes using the Scratch 3.0 program. A total of 292 Scratch projects were analyzed within the scope of the study. The frequency and corresponding percentage values of the criteria in the Scratch Projects Assessment Rubric showing how many of the projects were prepared according to the learning outcomes in the mathematics program at each secondary school grade level by 73 prospective teachers within the scope of the study are presented in Table 3.

Table 3. *Scratch projects assessment rubric results*

	5th grade		**6th grade**		**7th grade**		**8th grade**	
PROGRAMMING CONCEPTS	***f***	**%**	***f***	**%**	***f***	**%**	***f***	**%**
1. Sequence	67	91.8	67	91.8	70	95.5	71	97.3
2. User interaction	66	90.4	65	89.0	70	95.5	69	94.5
3. Iteration / Loop	44	60.3	51	69.9	54	74.0	65	89.0
4. Variables	36	49.3	32	43.8	39	53.4	48	65.8
5. Conditional statements	62	84.9	61	83.6	66	90.4	65	89.0
6. Lists (arrays)	2	2.7	3	4.1	4	5.5	4	5.5
7. Coordination and	40	54.8	41	56.2	55	75.3	56	76.7

synchronization								
8. Random numbers	19	26.0	12	16.4	13	17.8	11	15.1
9. Boolean logic	9	12.3	15	20.5	12	16.4	11	15.1
CODE ORGANIZATION								
10. Extraneous Blocks	20	27.4	16	21.9	8	11.0	6	8.2
11. Sprite names	7	9.6	8	11.0	10	13.7	12	16.4
12. Variable names	36	49.3	32	43.8	39	53.4	48	65.8
DESIGNING FOR USABILITY								
13. Functionality	49	67.1	57	78.1	63	86.3	65	89.0
14. Sprite customization	13	17.8	15	20.5	15	20.5	15	20.5
15. Stage customization	13	17.8	15	20.5	21	28.8	22	30.1
16. Clear instructions	61	83.6	62	84.9	63	86.3	65	89.0
17. App originality	21	28.8	32	43.8	30	41.1	40	54.8

According to table 3, when the Scratch projects prepared by the prospective secondary school mathematics teachers based on the learning outcomes in the mathematics program at each secondary school grade level are examined in terms of grade level, it is seen that *sequence*, *user interaction*, and *conditional statements* criteria in the Programming Concepts main category, the *extraneous blocks* criterion in the Code Organization main category and the *functionality* and *clear instructions* criteria in the Designing for Usability main category are provided in more projects at secondary school grade levels. The *lists*, *random numbers*, and *Boolean logic* criteria in the Programming Concepts main category, the *sprite names* criterion in the Code Organization main category, and the *sprite customization* and *stage customization* criteria in the Designing for Usability main category appear to be provided in fewer projects at all grade levels.

It is seen that many of the "Scratch Projects Evaluation Rubric" criteria of the projects prepared by prospective teachers for the 5th, 6th, 7th, and 8th-grade levels of secondary school have increased at the 8th-grade level compared to the 5th grade in case of usage. That is, the criteria are included in more of the last 8th-grade level projects compared to the 5th-grade level projects prepared at the beginning.

Accordingly, while the *iteration/loop*, *variables*, *coordination*, *and synchronization* criteria in the main category of Programming Concepts are used by the prospective teachers in more projects than at the beginning and the usage increase is higher, the usage case of the *random numbers* criterion is in fewer projects than at the beginning and the use of *boolean logic* and *lists (arrays)* criteria is not sufficient and its use appears to be low even though it is in a few more projects than the beginning.

Examining the main category of code organization, it is seen that the use of *extraneous blocks* in projects created by prospective teachers gradually decreases as more projects are created. As prospective teachers create more projects, it is seen that *variables* are used in more projects and this situation is also provided in the criterion of *giving meaningful names to the variables*. It is seen that the criterion of *sprite names* in the code organization category wasn't used adequately by prospective teachers, that is, they create their projects with the names in Scratch's own library for the sprites they use while creating their projects.

Examining the main category of designing for usability, it is seen that the *functionality* and *application originality* criteria are provided by the prospective teachers in more projects than at the beginning as more projects are created and the usage increase is higher, while the use of *sprite customization* and *stage customization* criteria is not sufficient even though they are provided in slightly more projects than the beginning.

Dr. Scratch Assessment Tool was used to analyze the proficiency level of Scratch projects related to 5th, 6th, 7th, and 8th-grade curriculum learning outcomes in terms of computational thinking concepts prepared by prospective teachers using the Scratch 3.0 program, and 292 Scratch projects were analyzed. The distribution of the obtained data is presented in Table 4.

Table 4. *Distribution of analysis results according to dr. scratch assessment tool*

	Basic		Developing		Proficiency		Total	
Grade Level	***f***	**%**	***f***	**%**	***f***	**%**	***f***	**%**
5th grade	6	8.2	42	57.5	25	34.2	73	100
6th grade	5	6.8	34	46.6	34	46.6	73	100
7th grade	3	4.1	35	47.9	35	47.9	73	100
8th grade	1	1.4	30	41.1	42	57.5	73	100

It is observed that 6 (8.2%) of the Scratch projects prepared by prospective teachers related to 5th-grade learning outcomes are at the basic level, 42 (57.5%) are at the developing level and 25 (34.2%) are at the proficiency level, 5 of the Scratch projects (6.8%) prepared for the 6th-grade learning outcomes are at the basic level, 34 of them (46.6%) are at the developing level and 34 (46.6%) are at the proficiency level, 3 of the Scratch projects prepared for the 7th-grade learning outcomes (4.1%) are at the basic level, 35 (47.9%) are at the developing level, 35 (47.9%) are at the proficiency level, 1 (1.4%) of the Scratch projects prepared for 8th-grade learning outcomes are at the basic level, 30 (41.1%) are at the developing level and 42 (57.5%) are at the proficiency level when Table 4 is examined. Accordingly, as the use of the Scratch program increases, it is seen that the adequacy level of the prepared projects is also increased.

The distribution of the average scores of the 7 computational thinking concepts in the Dr. Scratch assessment tool of the Scratch projects prepared by the prospective teachers according to each grade level is given in Table 5.

Table 5. *Score averages results of computational thinking concepts in dr. scratch assessment tool of projects prepared at grade level*

	Concepts of Computational Thinking						
Grade levels of prepared Scratch projects	Flow Control	Data Representation	Abstraction	User Interactivity	Synchronization	Parallelism	Logical Thinking
	$\overline{X}$	$\overline{X}$	$\overline{X}$	$\overline{X}$	$\overline{X}$	$\overline{X}$	$\overline{X}$
5th grade	2.03	1.53	1.12	1.95	2.34	2.10	1.75
6th grade	2.18	1.52	1.30	1.93	2.41	2.49	1.90
7th grade	2.29	1.60	1.11	1.97	2.53	2.53	1.92
8th grade	2.33	1.73	1.21	1.95	2.71	2.77	1.97

According to Table 5, when the average scores of the computational thinking concepts used by the prospective teachers in the Scratch projects they prepared for the 5th, 6th, 7th, and 8th-grade learning outcomes, respectively, are examined, the fact that the most common computational thinking concepts determined in the projects are flow control, synchronization, and parallelism, while the least is the abstraction is seen. In addition, the fact that the average scores of the computational thinking concepts used in the recent (related to the 8th-grade learning outcomes) projects of the prospective teachers are higher than the first projects they have done (related to the 5th-grade learning outcomes) is observed.

DISCUSSION, CONCLUSION, RECOMMENDATIONS

After the explanation and applications related to the Scratch program were completed in the scope of the research, the prospective teachers developed projects starting from the 5th-grade level related to the learning outcomes in the Mathematics Course Curriculum (Middle School 5th, 6th, 7th, and 8th Grades). A total of 292 projects, 73 at each grade level, were examined and evaluated. Accordingly, it was seen that most of the criteria in the "Scratch Projects Evaluation Rubric" were provided in more projects than at the beginning as new projects were developed by prospective teachers at different grade levels every week. It can be said that Scratch (Yoon et al., 2016), which is a good programming tool to teach programming concepts to new learners of programming, has a positive effect on the increase in prospective teachers' use of Scratch. Moreover, it helps to develop new projects (from 5th-grade level to 8th-grade level), to discover programming concepts and to reflect them in the projects.

Examining the projects made at each grade level, it is seen that the *sequence, user interaction,* and *conditional statements* criteria in the "Programming Concepts" main category of the Scratch Projects Assessment Rubric are met in most of the projects. Sequential execution (algorithmic order) is one of the basic concepts that a Scratch user encounters when starting to create a project (Maloney et al., 2008). In this sense, it is thought that most prospective teachers understand this basic programming concept from the moment they start creating Scratch projects. In the study conducted by Gabriele et al. (2019) with prospective teachers, it was seen that all projects prepared by prospective teachers provided *sequence* criteria. Similarly, in a study conducted by Öztürk (2021) with mathematics teachers, when the games designed by the teachers with the Scratch program were examined, it was seen that the *sequence* criterion was complied with in all the games.

Since the projects, demonstrate the use of *user interaction* are the majority at each grade level, indicated that the prospective teachers prefer to design projects in a way that requires the active use of the students and that the majority of their projects have been game-based. In addition, it is noted that most Scratch projects have the requirement of *user interaction* in the research examined in the projects in the literature (Gabriele et al., 2019, Maloney et al., 2008; Wilson et al., 2012).

In the main category of Programming Concepts, it has been seen that the criteria of *iteration/loops, variables, coordination, and synchronization* are included in more projects than at the beginning as more projects are created by prospective teachers and this situation is increasing. On the other hand, both at the beginning and as more projects are created, it becomes clear that programming concepts such as *random numbers*, *boolean logic*, and *lists(arrays)* are used in much fewer projects. The reason why these concepts are used in far fewer projects is that these programming concepts are difficult to learn (Gabriele et al., 2019) and not easily explored (Maloney et al., 2008).

Looking at the main category of Code Organization included in the rubric, the *extraneous blocks* usage errors that do not affect the functioning of the program (Gabriele et al., 2019) were seen less as the number of projects developed by prospective teachers increased. It is indicated that if prospective teachers are given the opportunity to develop projects more frequently, it may have a positive effect on the fact that such errors are less common. Similarly, studies examining Scratch projects in the literature have revealed that only a tiny percentage of students used excess blocks (Funke et al., 2017; Gabriele et al., 2019; Wilson, 2012).

Usability is when a tool can be used effectively for a specific purpose (Denner et al., 2012). It has been observed that as prospective teachers create more projects, they give more space to the *functionality* and *application originality* criteria in the main category of Designing for Usability. Therefore, as the experiences of prospective teachers increase, it can be said that the projects developed meet these criteria with an increasing trend. This positive change is considered to have created or will cause an increase in original projects. The projects developed by the prospective teachers were evaluated for the *Functionality* criterion, both in terms of suitability for the selected learning outcomes (suitable for the purpose) and the fact that the project was working without error. The *functionality* criterion also shows a positive course in the process. In direct proportion to the experiences of the prospective teachers, it is believed that there is a rise in the number of error-free projects appropriate for the chosen acquisition.

It was observed that the *Sprite names* in the Code Organization category and the *Sprite customization* and *Stage customization* criteria in the Designing for Usability category were included in very few of the projects developed and that there was not a sufficient increase in the use cases of the criteria in this category as the number of projects increased. In the study in which Funke et al. (2017) examined 127 Scratch projects made by university 4th-grade students, it was found that none of the sprites used in the developed projects had name editing, and most of the projects did not have sprite customization (85%) and stage customization (88%).

A total of 292 projects developed by 73 prospective teachers were evaluated by the Dr. Scratch assessment tool, an automated assessment tool. In the first stage, 6 of the projects prepared by the prospective teachers with 5th-grade learning outcomes were at the basic level and 25 of them were at the proficiency level, while 1 of the projects prepared with 8th-grade learning outcomes was at the basic level and 42 of them were at the proficiency level. It is seen that the average scores of the computational thinking concepts are higher in the last (8th grade) projects compared to the projects they first did for the 5th-grade level when the average scores of the projects according to the seven computational thinking concepts in the Dr. Scratch assessment tool are examined. The increased experience of prospective teachers using the Scratch software is assumed to be the cause of this positive change.

It was seen that as the prospective teachers gained experience when the programming concepts used in similarly developed projects were analyzed with the Scratch Projects Assessment Rubric, there was an increase in the number of projects in which many criteria in the rubric were used together. Therefore, it is true that developing more projects and gaining experience is effective in discovering both programming and computational thinking concepts.

The lowest average score was recorded in the concept of *abstraction*. In the study where the education process was evaluated by Papadakis and Kalogiannakis (2019) within the scope of Introduction to Programming with Scratch with prospective kindergarten teachers, 93 projects developed by prospective teachers at the end of the 13-week course were analyzed with Dr. Scratch assessment tool and it was seen that the lowest average score was in the concept of *abstraction*. Similarly, Hoover et al. (2016) evaluated computational thinking in students' game designs within the scope of the research given to 5 secondary school girls after the training was analyzed with the Dr. Scratch assessment tool and it was seen that the lowest score in the projects was generally for the concept of *abstraction*. Gabriele et al. (2019) also found that when they evaluated 40 Scratch projects made by prospective teachers with the Dr. Scratch evaluation tool, the concept of *abstraction* was used in the projects least. Troiano et al. (2020) used the Dr. Scratch assessment tool in their study to investigate how game types affect the development of computational thinking in Scratch games developed by 8th-grade students. As a result of the research, it was found that the concept of *abstraction* in most game types has low scores. In the literature, it is stated that it is difficult to teach *abstraction* to inexperienced novice users (Armoni, 2013). In this sense, although prospective teachers have developed projects at more than one different grade level in the process, it is thought that this is not enough for the development of the concept of *abstraction*.

The incorporation of technology into mathematics education helps to learn mathematical concepts in a meaningful way (İnce Muslu & Erduran, 2020). In this sense, programming can also be integrated into lessons, and teaching other topics through programming becomes more interesting, and it will be easier and faster for students to learn new concepts (Iskrenovic-Momcilovic, 2020). Depending on the results of this research, it is recommended for mathematics teachers and prospective mathematics teachers, to learn different block-based programs in order to integrate programming into their courses and to develop themselves in terms of programming concepts and computational thinking concepts in order to use the programs effectively. Apart from Scratch, it is recommended to conduct research with other block-based programming tools.

REFERENCES

Akkoç, H. (2013). Using technology in mathematics education for conceptual understanding. M.F. Özmantar, E. Bingölbali ve H. Akkoç (Ed). *Mathematical Misconceptions and Proposed Solutions* (pp. 361-392). Ankara: Pegem Academy.

Akpınar Y. & Altun, A. (2014). The need for programming education in information society schools. *Elementary Education Online 13*(1), 1-4.

Armoni, M. (2013). On teaching abstraction in CS to novices. *Journal of Computers in Mathematics and Science Teaching*, *32*(3), 265-284.

Atun, H. & Usta, E. (2019). The effects of programming education planned with TPACK framework on learning outcomes. *Participatory Educational Research, 6(2), 26-36.*

Aytekin, A., Sönmez Çakır, F., Yücel, Y.B. & Kulaözü, İ. (2018). Coiıng science directed to future and some methods to be available and coding learned. *Eurasian Journal of Researches in Social and Economics (EJRSE), 5*(5), 24-41.

Calder, N. (2018). Using Scratch to facilitate mathematical thinking'. *Waikato Journal of Education, 23*(2), 43-58. https://doi.org/10.15663/wje.v23i2.654.

Demir, Ö. & Seferoğlu, S.S. (2017). New concepts, different uses: An assessment of computational thinking. H.F. Odabaşı, B. Akkoyunlu & A. İşman (Ed.). *Educational technology readings 2017* (pp. 801-830). Sakarya: Sakarya University Press.

Denner, J., Werner, L., & Ortiz, E. (2012). Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts? *Computers &Education, 58*(1), 240–249. https://doi.org/10.1016/j.compedu.2011.08.006.

Doğan, S. (2014). Functions of education. C.T. Uğurlu (Ed.), *Introduction to educational science* (pp.39-53). Eğiten Book.

Erdoğan, T. & Şimşek, A. (2018). Teaching programming and drama. Y. Gülbahar & H. Karal (Ed.), *Teaching programming from theory to practice* (pp. 237-269). Ankara: Pegem Academy.

Ersoy, Y. (1997). Mathematics education in schools: Literacy in mathematics. *Hacettepe University Journal of Education, 13*, 115-120.

Erümit, A. K. & Berigel, M. (2018). History of programming languages and programming teaching. Y. Gülbahar ve H. Karal (Ed.), *Teaching programming from theory to practice* (pp. 2-36). Ankara: Pegem Academy.

Funke, A., Geldreich, K., & Hubwieser, P. (2017, April). Analysis of scratch projects of an introductory programming course for primary school students. *Global Engineering Education Conference (EDUCON)*,1229-1236.

Gabriele, L., Bertacchini, F., Tavernise, A., Vaca-Cárdenas, L., Pantano, P., & Bilotta, E. (2019). Lesson planning by computational thinking skills in Italian pre-service teachers. *Informatics in Education, 18*(1), 69-104.

Genç, Z. & Karakuş, S. (2011). Learning through design: Using scratch in instructional computer games design. *5th International Computer & Instructional Technologies Symposium, 22-24 September 2011 Fırat University*, Elazıg, Turkey.

Germia, E. & Panorkou, N. (2020). Using scratch programming to explore coordinates. *Mathematics teacher: Learning & Teaching PK-12 113* (4), 293-300. https://doi.org/10.5951/MTLT.2018.0032.

Gomes, A. & Mendes, A. (2007). Learning to program difficulties and solutions. *International Conference on Engineering Education ICEE 2007*. Coimbra, Portugal.

Güleryüz, H., Dilber, R. & Erdoğan, İ. (2020). Prospective Teachers' Views on Coding Training in Stem Applications. *Journal of Ağrı İbrahim Çeçen University Social Sciences Institute, 6* (1), 71-83. https://doi.org/10.31463/aicusbed.610909.

Hoover, A. K., Barnes, J., Fatehi, B., Moreno-León, J., Puttick, G., Tucker-Raymond, E., & Harteveld, C. (2016, October). Assessing computational thinking in students' game designs. In *Proceedings of the 2016 annual symposium on computer-human interaction in play companion extended abstracts* (pp. 173-179).

Iskrenovic-Momcilovic, O. (2020). Improving geometry teaching with scratch. *International Electronic Journal of Mathematics Education*, *15*(2). https://doi.org/10.29333/iejme/7807

İnce Muslu, B. & Erduran, A. (2020). Analyzing of Technology Integration Process in Mathematics Education. *The Journal of Buca Faculty of Education*, *50*,258-273.

Karal, H., Çakmak – Şılbır, G. & Yıldız, M. (2018). The role of computational thinking and coding in STEM education. S. Çepni (Ed.). *STEM education from theory to practice* (pp. 397-419). Ankara: Pegem Academy.

Lewis, C. M. & Shah, N. (2012). Building upon and enriching grade four mathematics standards with programming curriculum. *43rd ACM technical symposium on Computer Science Education*, 57–62. https://doi.org/10.1145/2157136.2157156.

Maloney, J. H., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. (2008, March). Programming by choice: urban youth learning programming with scratch. In *Proceedings of the 39th SIGCSE technical symposium on Computer science education*, 367-371.

Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The scratch programming language and environment. *ACM Transactions on Computing Education,* 10(4), 1-16. https://doi.org/10.1145/1352135.1352260.

Miles, M.B. & Huberman, A.M. (1994). Qualitative Data Analysis: An Expanded Sourcebook (2nd ed.). Thousand Oaks, California: SAGE.

Ministry of National Education [MONE]. (2018). Secondary School Mathematics (Grades 9, 10, 11 and 12) Curriculum Retrived September 10, 2021, from http://mufredat.meb.gov.tr/Dosyalar/201821102727101OGM%20MATEMAT%C4%B0K%20PRG%2020.01.2018.pdf.

Mo, W., Zhang, Y., Fu, Y., Lin, J., Gao, H. & Lin, Y. (2021). Effects of a scratch-based extenential learning approach on students' math learning achievements and interest. 2021 International Symposium on Educational Technology (ISET), 261-265. 10.1109/ISET52350.2021.00062.

Moreno-Leon, J. & Robles, G. (2015). *Analyze your Scratch projects with Dr. Scratch and assess your computational thinking skills*. *Scratch2015Ams*.

Moreno-León, J., Robles, G. & Roman- Gonzalez, M. (2015). Dr. Scratch: Automatic analysis of scratch projects to assess and foster computational thinking. *Revista de Educación a Distancia 46*, 1-23.

Oluk, A. & Çakır, R. (2021). The effect of code.org activities on computational thinking and algorithm development skills. *Journal of Teacher Educational and Lifelong Learning,3*(2), 32-40.

Öztürk, A. (2021). *Examining the games designed by secondary school mathematics teachers with the Scratch program in line with the teacher and student views: Reflections from algebra* (Master's thesis). Bartın University.

Papadakis, S., & Kalogiannakis, M. (2019). Evaluating a course for teaching İntroductory programming with Scratch to pre-service kindergarten teachers. *International Journal of Technology Enhanced Learning*, *11*(3), 231-246.

Setyawan, T. Y. (2020). Primary school pre-service teachers' competence level of computational concepts in programming using Dr. Scratch. *Jurnal Inovasi Teknologi Pendidikan*, *7*(2), 177-187. https://doi.org/10.21831/jitp.v7i2.36185

Şimşek, E. (2018). *The effect of robotics and scratch applications on computational thinking skills and* *academic achievement in programming instruction* (Master's thesis). Ondokuz Mayıs University.

Troiano, G. M., Chen, Q., Alba, Á. V., Robles, G., Smith, G., Cassidy, M., Tucker- Raymond, E., Puttick, G. & Harteveld, C. (2020, April). Exploring How Game Genre in Student-Designed Games Influences Computational Thinking Development. *2020 CHI Conference on Human Factors in Computing Systems*, 1-17. https://doi.org/10.1145/3313831.3376755.

Usta, E. & Korkmaz, Ö. (2010). Pre-service teachers' computer competencies, perception of technology use and attitudes toward teaching career. *International Journal of Human Sciences, 7*(1), 1335-1349.

Üzümcü, Ö. & Bay, E. (2018). A new 21st century skill in education: Computational thinking *Uluslararası Türk Kültür Coğrafyasında Sosyal Bilimler Dergisi, 3*(2), 1-16.

Wilson, A., Hainey, T., & Connolly, T. (2012, October). Evaluation of computer games developed by primary school children to gauge understanding of programming concepts. *6th European Conference on Games Based Learning*, 549.

Wing, M. J. (2006). Computational thinking. *Communications Of The ACM 49*(3), 33-35. https://doi.org/10.1145/1118178.1118215.

Yıldırım, A. & Şimşek, H. (2006). Qualitative Research Methods. Ankara: Seçkin Publishing.

Yıldız Durak, H. and Karaoğlan Yılmaz, F. G. (2019). An Investigation of Prospective Teachers' Educational Digital Game Designs for Mathematics Teaching and Their Opinions on the Design Process. *Ege Journal of Education, 20*(1), 262-278. https://doi.org/10.12984/egeefd.1095321.

Yıldız Durak, H. Karaoğlan Yılmaz, F. G. & Yılmaz, R. (2018). Examination of Educational Project Products Prepared by Teacher Candidates During Programming Instruction With Scratch. *International Symposium of Academic Studies on Education and Culture Full Text Paper Book*, 129-136.

Yoon, I., Kim, J., & Lee, W. (2016). The analysis and application of an educational programming language (RUR-PLE) for a pre-introductory computer science course. *Cluster Computing*, *19*(1), 529-546.

Yükseltürk, E. & Üçgül, M. (2018). Block based programming. Y. Gülbahar ve H. Karal (Ed.), *Teaching programming from theory to practice* (pp. 2-36). Ankara: Pegem Academy.