ISSN: 2149-214X

Journal of Education in Science, Environment and Health

www.jeseh.net

An Investigation of Coding Education Practices in terms of Primary School Students' Algorithmic Thinking Skills and Students' Opinions

Anil Erkan[1], Sumeyra Akkaya[2]
[1] Toki Firat Primary School
[2] Inonu University

To cite this article:

Erkan, A., & Akkaya, S. (2025). An investigation of coding education practices in terms of primary school students' algorithmic thinking skills and student's opinions. *Journal of Education in Science, Environment and Health (JESEH), 11*(1), 47-73. https://doi.org/10.55549/jeseh.769

Journal of Education in Science, Environment and Health
Volume 11, Issue 1, 2025 https://doi.org/10.55549/jeseh.769

An Investigation of Coding Education Practices in terms of Primary School Students' Algorithmic Thinking Skills and Students' Opinions

Anil Erkan, Sumeyra Akkaya

Article Info

Article History

Published:
01 January 2025

Received:
15 September 2024

Accepted:
30 December 2024

Keywords

Coding education,
Scratch program,
Algorithmic thinking skills

Abstract

This study aims to examine the views of fourth-grade primary school students on coding education given through the Scratch program by determining the students' skills in using the program and algorithmic thinking skills. The study was conducted as a one-group study with an embedded mixed design. The study group consisted of 32 students attending the 4th grade in a primary school. The data were collected using the Algorithmic Thinking Skills Scale developed by the researcher, Semi-structured Interview Form, Student Diaries, Scratch Programming Skills Checklists and Researcher Diary. According to the quantitative results of the study, a significant difference was found between the Algorithmic Thinking Skill Scale pre-test and post-test results of the students in the experimental group. According to the results obtained from the Scratch checklists among the qualitative results of the study, it was observed that the students in the experimental group generally added the decor, characters and codes in the projects correctly. According to the student diaries among the qualitative results of the study, it was determined that students generally found the coding course and the projects done in the course enjoyable. The data obtained from the researcher's diaries also supported the student diaries. Finally, according to the student opinions among the qualitative results of the study, it was concluded that the students liked the Scratch program enjoyed making projects and did not get bored. In this context, it is recommended to provide coding education to primary school students and to teach the Scratch program to primary school students.

Introduction

Coding

In recent years, coding education has become increasingly popular, especially among young children. From kindergarten to high school, coding education or programming is implemented in both public and private schools. Teaching coding or programming to children has become more accessible with the widespread availability of coding tools that make programming more engaging during the teaching process (Erol, 2020). Coding involves the step-by-step creation of instructions that outline the processes required for a specific action to be performed using computers. In other words, it is the process of developing various solutions to existing problems by using a language that computers can understand (Yigit, 2016). It also entails the sequential execution of predefined commands and the clear articulation of tasks to be performed through computers in the form of instructions (EBA, 2024). Coding is the process of writing one or more commands on a computer system to perform a specific task. It involves converting an algorithm developed to achieve a goal into a programming language. Computer programming or coding can be explained as the development and implementation process that uses various command sets to accomplish predetermined tasks or operations on a computer, solve encountered problems, and establish the necessary interaction between humans and computers (Sayın & Seferoğlu, 2016). Research indicates that coding education enhances children’s high-level thinking skills, such as approaching problems from different perspectives, thinking systematically, generating solutions, engaging in creative thinking, and establishing cause-effect relationships (Yukselturk & Altıok, 2016). Furthermore, coding education helps students develop essential skills, including analytical thinking, creativity, digital literacy, problem-solving, collaborative work and learning, process- and result-oriented thinking, spatial reasoning, and learning through hands-on experiences (Akpınar & Altun, 2014; Demirer & Sak, 2016). Children who acquire coding skills at an early age also gain experience in communication, critical thinking, and problem-solving, which are crucial for developing 21st-century skills. These skills are indispensable for the future success of our children in an increasingly digital world (Mclennan, 2017).

At the core of coding lies algorithms. In other words, knowing how to create algorithms is essential for coding. Through coding education, students are taught algorithms and are expected to use these algorithms to write codes. According to Calıskan (2020), students who receive coding education gain numerous learning opportunities, such as the ability to construct algorithms, code the created algorithms, program effectively, develop different perspectives, and use shortcuts in problem-solving.

Algorithm

An algorithm is the sequential representation of instructions required to solve a problem within a logical framework (Arı, 2016; Ercil- Cagıltay & Fal, 2016). It is a structure composed of steps that must be followed to achieve a specific goal or solve a problem (Aytekin et al., 2018; Gibson, 2012). Olsen (2000) defined an algorithm as a set of instructions designed to solve a problem.

Algorithms are not solely a phenomenon related to computers. Much like the universal language of music that resonates with the human soul, algorithms serve as a universal language for problem-solving. Although we may not always realize it, we employ algorithms in our daily lives. In other words, algorithms are akin to computational geometries that enable individuals to articulate their thoughts. They offer general solutions to problems and play a critical role in addressing everyday challenges and contributing to advancements in the technological era (Arı, 2016; Aytekin et al., 2018). Writing an algorithm is a meticulous process that demands careful attention, as every step needs to be planned in detail. An error in any step can directly impact the outcome. Therefore, certain rules must be followed when creating an algorithm. Kucukkoc (2020) outlined these rules as follows:

- All lines in the algorithm should be numbered starting from 1.
- The first line of every algorithm should begin with "1. Start."
- Every algorithm must conclude with the "Stop" command.
- Process flow directions within the algorithm should be provided using the "Git" instruction along with the line number.
- Functions or subroutines in algorithms should be named clearly with their identifying parameters.
- The steps in the algorithm should be clear, limited in number, precise, and unambiguous.
- The expressions in the algorithm should be as simple and comprehensible as possible.
- The instructions in the algorithm should not rely on any specific operating system, hardware, or programming language.

Example Algorithm

The algorithm for multiplying two numbers (e.g., 4 × 3 = 12) can be represented as shown in Figure 1. An algorithm is not just a tool for problem-solving; it encompasses the thinking processes through which individuals develop important skills to organize and analyze their thought patterns. In this context, algorithms are not only instructions that perform specific tasks but also tools that shape individuals' ways of thinking. The correct writing and application of an algorithm demonstrate the importance of algorithmic thinking skills. Algorithmic thinking is not just about understanding how algorithms work, but also the ability to use these processes effectively.

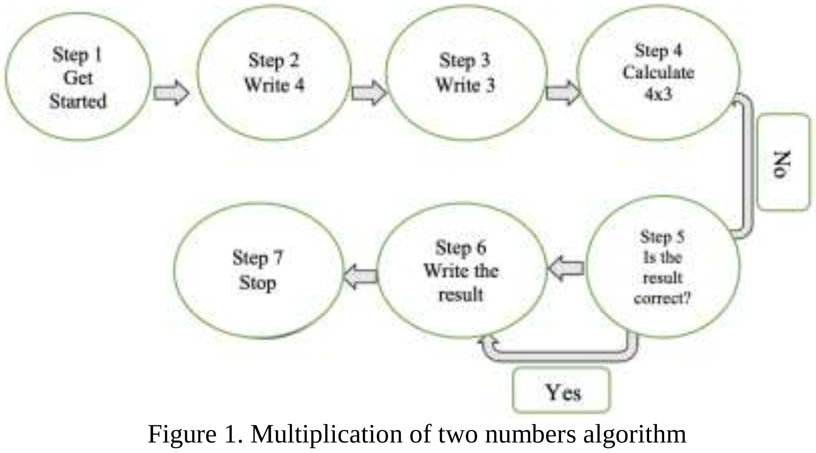


Figure 1. Multiplication of two numbers algorithm

Algorithmic Thinking Skills

According to Brown (2015), algorithmic thinking is the ability to understand, create, apply, and evaluate algorithms. Csizmadia et al. (2015) define algorithmic thinking as the ability to clearly define the steps to be taken when faced with a problem and the path to follow to reach a solution. Rather than finding a single solution to a problem, algorithmic thinking involves developing rules and instructions that can generate solutions not only for the current situation but also for similar problems, by creating various algorithms. The strength of algorithmic thinking lies in making these solutions practical. The ability to think in terms of rules and sequences to understand encountered situations or solve problems is also a key component of algorithmic thinking. The programming language used to teach algorithmic thinking skills is not sufficient for algorithmic thinking design. This is because, when teaching a programming language, too much emphasis is placed on the features of the language, and insufficient time is allocated to developing algorithmic thinking. Therefore, it would be more effective to create algorithms using code that is appropriate to the student's level, rather than focusing solely on a programming language when teaching algorithmic thinking (Futschek, 2006).

Scratch

Scratch is a programming tool that teaches programming to users between the ages of 8 and 16 while they work on projects such as stories, games, and animations (Maloney et al., 2010). The Scratch program, marketed with the slogan "Imagine, code, share," was developed by MIT's (Massachusetts Institute of Technology) Media Lab. The program is used in over 150 countries and is available in more than 40 languages. Scratch, which allows code blocks to be connected through logical steps, can be described as a program that enables real-world problem-solving (Kızılkaya & Sart, 2017). Scratch is a coding program that allows individuals or groups to design activities with code blocks in an environment that incorporates games, and then share these designed activities with other users (Scratch About, 2023). Scratch, a programming language, is the first programming tool many people learn. It is specifically designed for beginners. Scratch is an ideal program for creating games and animations, and for teaching coding while creating them. It offers both ease of use and speed. Dragging and combining pre-made code blocks in the program facilitates project creation (Dickins et al., 2016). It is easier to produce projects by combining code blocks in Scratch than with other text-based programming projects (Genc & Karakus, 2011). It has been noted that while programming projects using Scratch are both created and shared, 21st-century skills such as systematic thinking, collaborative thinking, and creative thinking are developed (Resnick et al., 2009). Additionally, since the Scratch program is game-based, lessons can become more enjoyable, motivation can increase, and students' creative thinking and problem-solving skills can be enhanced (Gezgin et al., 2017).

Because Scratch has a visual and flexible structure, it removes the barriers to programming and enables young people to create games and develop animations. Scratch's stage is similar to a real-world theater stage. It allows especially talented students to unleash and reflect on their imaginations. The software also teaches students mathematical elements, such as the coordinate system (Lee, 2011). Scratch is a tool that students can use to learn and enjoy programming, thanks to its interactive structure and fun characters. Scratch, a programming language, is a platform that can be used by both children and adults to learn coding. No additional software is needed to use Scratch, as it also works through web browsers (Demirkol, 2017). Scratch is a wonderful programming tool that allows students to design individual animations, and games, or create interactive stories using rich media tools like sound, music, and images, all within a pleasant environment, and share these projects (Cubukluoz, 2019; Yukselturk & Ucgul, 2018). Scratch makes programming fun and helps students easily learn basic algorithm concepts and programming skills (Yukselturk & Ucgul, 2018).

Scratch helps students develop creative thinking, logical reasoning, problem-solving, and collaboration skills, and contributes to their learning of computer and mathematical concepts (Su, 2019). By using Scratch, students can improve their problem-solving skills, propose different solutions to problems, and design creative activities. They can also gain experience by working on projects, participating in collaborative learning activities, and redesigning their projects (Kordaki, 2012; Resnick et al., 2009). This study aims to fill a significant gap in coding education for elementary school students. In today's world, the widespread use of technology and the necessity of acquiring digital skills from an early age have increased the importance of coding education. As a visual programming language, Scratch enables children to develop their algorithmic thinking skills, enhance their problem-solving abilities, and foster creative thinking. Therefore, investigating the feasibility of implementing coding education at earlier ages using the Scratch program will contribute to helping students adapt to technology and meet the demands of the digital age. The contributions of this study to the literature can be summarized as follows:

- *Early coding education*: By exploring the feasibility of implementing coding education, typically offered at the middle school level, for elementary school students, the study will provide scientific data to address this gap.
- *Algorithmic thinking skills*: The study will reveal the impact of Scratch on students' algorithmic thinking skills, offering evidence that these skills can be developed at an early age.
- *Educational practices*: By analyzing students’ perceptions and experiences during coding lessons using Scratch, the study will guide educators on how to use such programs more effectively.
- *Teacher perspectives*: The study will examine teachers' experiences and opinions on Scratch-based coding education, offering suggestions for integrating such practices into the education system.
- *Designing their own gam*es: By investigating students' processes of designing their own games after Scratch training, the study will highlight gains such as creative productivity and self-confidence.

This study not only demonstrates the benefits of early coding education but also serves as an essential resource for expanding coding education by contributing to education policies and curriculum development processes. The study aims to examine students' skills in using the Scratch program, their algorithmic thinking skills, and their perspectives on coding education provided through Scratch. It was conducted to determine whether coding education, typically offered at the middle school level, can be implemented at an earlier age. By teaching the Scratch program to elementary school students, the aim is to develop their coding skills, enhance their thinking abilities, teach the fundamental logic of coding, and enable them to design their games by the end of the training.

1. Does the Scratch program affect students' algorithmic thinking skills?
2. What are the students' abilities to use the Scratch program?
3. What are the students' perceptions and experiences regarding the lessons conducted using the Scratch program?
4. What are the teacher's perceptions and experiences regarding the lessons conducted using the Scratch program?
5. What are the students' opinions about coding education with Scratch?

Method

Research Design

To make the research more comprehensive, a mixed methods approach, which involves combining qualitative and quantitative data, was used. According to Johnson, Onwuegbuzie, and Turner (2007), mixed methods research is when a researcher conducts comprehensive research by integrating both qualitative and quantitative research approaches, collecting data, analyzing it, and drawing inferences. In this mixed-method study, in which we collected and analyzed both qualitative and quantitative data together, the embedded design, one of the mixed-methods designs, was used. Embedded design is a mixed methods approach in which the researcher combines traditional quantitative or qualitative data and analyzes both types of data. In an embedded design, the researcher may add a qualitative phase to a quantitative study using an experimental design, or a quantitative phase to a qualitative study using a case study. In embedded design, the phase used as support is intended to enhance the overall design (Delice, 2018; Gultekin et al., 2020). A weak experimental design was used in the quantitative dimension of the research. Among the experimental design types, a single-group pretest-posttest design was used, while a phenomenological research design was used in the qualitative dimension. The single-group pretest-posttest design is an experimental design conducted on an experimental group, which examines the effect of the independent variable through pretest and posttest measurements.

Since there is no control group in this design, it is considered a weak experimental design (Creswell & Plano-Clark, 2018). A phenomenological research design was also used in the qualitative dimension of the study. In phenomenological studies, the experiences of individuals related to the events or situations they have encountered are typically examined. The researcher conducts interviews with individuals to uncover their experiences with these events, and by analyzing the data obtained from these interviews, the researcher defines the phenomena.

Study Group

The study group consists of 32 students from the 4th grade at a school in the central district of Elazığ province during the 2022-2023 academic year. The study was conducted with a single group. The study group was selected using convenience sampling, a method within purposive sampling. Convenience sampling is defined as selecting the sample from easily accessible and applicable units, providing speed and practicality to the study (Buyukozturk et al., 2020).

Data Collection Tools

In the study, the Algorithmic Thinking Skills Scale developed by the researcher, along with the Scratch Programming Skills Checklists, Student Diaries, Researcher Diary, and Semi-Structured Interview Form, all prepared by the researcher, were used. The reliability coefficient ($\alpha = .89$) of the 'Algorithmic Thinking Skills Scale' developed by the researcher. Considering the reliability values table explained by Özdamar (2017), this coefficient is considered 'excellent in terms of reliability.' The Scratch programming checklist prepared by the researcher was created for each project taught and used individually for each student. Each step of the checklist consists of the codes found in the Scratch project. All the codes are listed in detail in the checklist. The checklist includes a column where the researcher needs to mark whether each student has executed the codes correctly or incorrectly. This allows the researcher to indicate which student made which code correct or incorrect in the relevant section. In the student diaries prepared by the researcher, students wrote their feelings, thoughts, and experiences related to the lesson of the day at the end of each class and submitted them to the researcher. In the researcher's diary, the researcher observed the process and recorded these observations to create the researcher's diary. During the process, students' attitudes and behaviors were examined in detail, and notes were taken. These notes were compiled, and the final version of the researcher's diary was created.

Implementation Process

The implementation phase of the study lasted 10 weeks (20 class hours). In the first 2 weeks (4 class hours), the Scratch program was introduced to the students. The code blocks in the program were explained in detail, separately. Game examples were shown, and students were allowed to play these games. In the remaining 8 weeks (16 class hours), games were designed with the students. Activities were planned as one game per week. In this way, students were taught 8 game design methods and were asked to design these games individually. The implementation process was carried out by the researcher, who guided the students at every stage of the process and explained the points they found difficult or could not understand.

The researcher created a teacher account on the https://scratch.mit.edu website. Then, she created a class through this account and included the students in the class. The researcher created a studio for the game designed each week and asked the students to upload their projects to this studio. Finally, the researcher examined the uploaded projects one by one and ensured that any deficiencies were addressed by notifying the students via the system. The implementation stages, the designed games, and the implementation period in the experimental group during the implementation process are shown in Table 1.

Table 1. Experimental group implementation process

Week	Implementation Phase	Implementation Duration
Week 1	Algorithm and Programming	80 min
Week 2	Introduction of Scratch Program	80 min
Week 3	Dino-Dog Project	80 min
Week 4	Ball Bouncing Project	80 min
Week 5	Star Catching Project	80 min
Week 6	Labyrinth Project	80 min
Week 7	Flying Cat Project	80 min
Week 8	Color Capture Project	80 min
Week 9	Wheel of Fortune Project	80 min
Week 10	Jumping the Barrier Project	80 min

Data Analysis

In this mixed-method study, quantitative and qualitative data were analyzed separately. Techniques such as standard deviation (SD), frequency, and percentage distributions were used to analyze the quantitative data. Before analyzing the data of the experimental group, it was checked whether the data followed a normal distribution. If the statistically insignificant value (p-value) is above .05, it indicates that the data follow a normal distribution (Pallant, 2017). The normality data for the pretest and post-test scores of the Algorithmic Thinking Skills Scale of the experimental group are provided in the tables below. When the normality value of the Algorithmic Thinking Skills Scale pretest scores of the experimental group was examined, the p-value of significance was .361 (Table 2). This value supports the assumption of normality. According to the normality data, the pretest results of the experimental group show a normal distribution.

Table 2. Analysis results regarding the normality distribution of the pretest scores of the experimental group

	Shapiro-Wilk		
Experimental Group Pre-Test	Statistics	Sd	p
Total	.964	32	.361

Table 3. Analysis results regarding the normality distribution of the posttest scores of the experimental group

	Shapiro-Wilk		
Experimental Group Posttest	Statistics	Sd	p
Total	.721	32	.000

When the normality value of the Algorithmic Thinking Skills Scale posttest scores of the experimental group was examined, the p-value of significance was .000 (Table 3). This value violates the assumption of normality. According to the normality data, the post-test results of the experimental group do not show a normal distribution. Content analysis was used to analyze the qualitative data. Student diaries and semi-structured interview forms were coded based on the feedback received, and main themes were developed. The process was followed using checklists prepared by the researcher and supported by the researcher's diaries. The coding process used to analyze qualitative data was carried out by the researchers who prepared the transcripts of the interviews. Then, the main themes and categories relevant to the research questions were identified, and important statements in the interview data were labeled and coded according to these themes. During the coding process, main themes were divided into subcategories when necessary. The coded data was analyzed, similar statements were grouped, and relationships were established. This process ensured that the interview data were examined in a more systematic and meaningful way.

Findings

Findings Related to Quantitative Data

Wilcoxon Signed Ranks Test of Algorithmic Thinking Skill Scale Pre-Test and Post-Test Scores of the Experimental Group

It was examined whether the scores obtained from the Algorithmic Thinking Skills Scale, administered to the experimental group students before and after the application, showed a statistically significant difference. The data from the Wilcoxon Signed Ranks Test conducted for this purpose are presented in Table 4.

Table 4. Wilcoxon signed ranks test for experimental group pre-test - post-test scores

Experimental Group Pre-Test - Post Test	N	Rank Mean	Row Totals	z	p	Effect Size
Negative Sequence	7	10.14	71.00			
Positive Sequence	23	17.13	394.00	-3.331	.001	.588
Equal	2					

When Table 4 was examined, it was observed that the difference between the post-test and pre-test scores of 7 students in the experimental group was negative, the difference between the post-test and pre-test scores of 23 students was positive, and the difference between the post-test and pre-test scores of 2 students was zero. According to the data obtained, when the pre-test and post-test results of the Algorithmic Thinking Skills Scale for the experimental group students were analyzed, the z-value was found to be -3.332 and the p-value was .001 (p-value .000 < .05). Based on this, it was concluded that there was a significant difference between the pretest and posttest scores of the experimental group students' algorithmic thinking skills, favoring the posttest score. The value (r = .588) obtained from the pretest and posttest scores of the experimental group students indicates a large effect size.

Findings Related to Qualitative Data

Findings Related to Algorithm and Programming

The data of the student diaries applied to the students after the Algorithm and Programming course are given in Table 5. Since the students pointed to more than one category while expressing their opinions in their diaries, the number of opinions in the categories was higher than the number of students in the experimental group. When Table 5 was examined, it was observed that 48 of the experimental group students' feelings, thoughts, and experiences about the Algorithm and Programming course were collected in two categories: “Positive” (48 opinions) and “Negative” (8 opinions). Under the positive category, codes were grouped into 5 subheadings, and under the negative category, codes were grouped into 2 subheadings.

Table 5. Student diaries on the introduction of algorithms and programming

Category / Code		f
Positive		48
	It was a lot of fun	15
	I'm so excited	12
	It was very nice to command	10
	It was fun to execute commands	6
	It was very striking.	5
Negative		8
	At first, I didn't know what to do	5
	I was afraid of not being able to say the right commands	3

When the opinions in the positive category were analyzed in detail in the table, it was found that 15 students expressed that the lesson was very fun, 12 students were very excited, 10 students enjoyed giving commands, 6 students found it enjoyable to apply the commands, and 5 students stated that the lesson was very remarkable. When the opinions in the negative category were analyzed in detail, 5 students expressed that they did not know what to do at first, and 3 students were afraid of not being able to give the correct commands.

S2: "The lesson was very fun. Especially giving commands was very enjoyable."
S5: "I was very excited when our teacher came to the lesson with a toothbrush and toothpaste. When we started the lesson, I did not know what to do at first. I thought what if I gave the commands wrong. But it was not as I feared, I had a lot of fun."
S10: "The lesson was very fun. I enjoyed giving commands and practicing the given commands."

The researcher's diary of the Algorithm and Programming course contains the following statements:

"When I entered the classroom with a toothbrush and toothpaste, the students paid attention. It was clear from their eyes that they were wondering what I was going to do with what I had. When I started to explain the lesson, they were listening to me with curious eyes. They were very surprised when they learned what an algorithm was. They loved realizing that the actions they do in daily life are algorithms. They were very impatient to stand up at the blackboard and give commands and follow them. They were very happy at the end of the lesson."

Findings Related to the Introduction of Scratch Program

The data of the Student Diaries applied to the students after the Introduction of Scratch Program course are given in Table 6. When Table 6 was examined, it was observed that 53 of the experimental group students' feelings, thoughts, and experiences about the Scratch Program were gathered under three categories: “As an Entertainment Tool” (f=53), “From an Educational Perspective” (f=17), and “From a Time Perspective” (f=3). Under the category "As an Entertainment Tool," the codes were grouped into 5 subheadings; under the category "From an Educational Perspective," the codes were grouped into 3 subheadings; and under the category "In Terms of Time," the codes were grouped into 2 subheadings. When the opinions in the "As an Entertainment Tool" category were examined in detail, 21 students expressed that it was very entertaining, 16 students described it as very beautiful, 8 students said it was very loved, 7 students were very excited, and 1 student stated that the lesson was very special. When the opinions in the "From an Educational Perspective" category were examined in detail, 8 students expressed that they learned how to make games, 7 students thought that the program was educational, and 2 students considered it a good program to improve themselves. When the opinions in the "In Terms of Time" category were examined in detail, 2 students expressed that it was a bit challenging at first, and 1 student mentioned that they could not fully understand some of the codes.

S7: "Scratch program is very nice. It is very fun to make games. It is very enjoyable to write the codes."
S15: "I was very excited when I met the Scratch program. The program is very fun. I am very excited to make games with this program. I am glad I met this program."
S20: "It is a very nice program, but I still don't understand some codes. I don't know how to do them, but it is a very good program to improve ourselves."

Table 6. Student diaries on the introduction of the Scratch program

Category / Code		f
As a means of entertainment		53
	It's a lot of fun	21
	It was very beautiful	16
	I like it a lot	8
	I'm so excited	7
	It was very special	1
Educational aspects		17
	Learning to make games	8
	Educational	7
	Good for self-improvement	2
In terms of time		3
	A Little Tricky	2
	I don't fully understand some codes	1

The following statements are also included in the researcher's diary of the Introducing Scratch Program course:

"The students seem to enjoy going to the computer lab. I observed that they were very excited when we introduced the Scratch Program today. The main page of the program and the code sections attracted the attention of the children. They constantly asked questions about the program. There are so many things they are curious about. They want to learn everything in one day. The lessons look like they will be very enjoyable.''

Table 7. Checklist for the Dino-Puppy Project

Codes	Added by (f)	Did not add (f)
Decorations added 3 scenes	31	1
Dog 1 added his puppet	31	1
Added Dinosaur 4 puppet	28	4
Bat added his puppet	32	-
For the bat puppet  added the code.	32	-
For the Dinosaur 4 puppet  added the code	32	-
For Dog 1 puppet  added the code	32	-

Findings of the Dino-Dog Project

Data on the checklist applied for the Dino-Dog Project are given in Table 7. When the checklist of the Dino-Dog Project in Table 7 was examined, it was observed that 31 students in the experimental group added decorations to 3 scenes, 1 student affected a different scene, 31 students added the Dog 1 puppet, 1 student added a different puppet, 28 students added the Dinosaur 4 puppet, 4 students added a different puppet, and all students added the Bat puppet. When the code blocks were analyzed, it was found that all students wrote the correct codes for the Bat, Dinosaur 4, and Dog 1 puppets. Analyzing the data in the Movements Checklist revealed that all students added the codes correctly, and only a few students used different props and puppets in the scenes.

Table 8. Student diaries related to the Dino-Dog Project

Category / Code		f
As a means of entertainment		54
	It was beautiful	28
	It was a lot of fun	20
	New projects excite me	3
	I lost track of time	2
	I want to code every day	1
	It's fun to watch my project	1
	The characters were chirpy	1
Educational aspects		6
	Educational	3
	I learned codes with the project	1
	Looks like an algorithm	1
	I made small mistakes at the beginning	1
In terms of time		3
	It didn't take much effort	2
	It took some time to write the code	1

The data of the Student Diaries applied to the students after the Dino-Dog Project lesson are given in Table 8. When Table 8 is examined, it is observed that 54 of the experimental group students' feelings, thoughts, and experiences about the Dino-Foam project were categorized under three headings: “As a Means of Entertainment” (f=54), “From an Educational Perspective” (f=6), and “From a Time Perspective” (f=3). Under the category "As a Means of Entertainment," the codes were grouped into 7 subheadings; under the category "From an Educational Perspective," the codes were grouped into 4 subheadings; and under the category "In Terms of Time," the codes were grouped into 2 subheadings. When the opinions under the "As a Means of Entertainment" category were examined in detail, 28 students expressed that it was very beautiful, 20 students said it was very entertaining, 3 students were excited to do new projects, 2 students did not understand how time passed, 1 student wanted to code every day, 1 student enjoyed watching the project, and 1 student thought the characters were chirpy. When the opinions in the "Educational Perspective" category were analyzed in detail, 3 students stated that it was educational, 1 student learned the codes with the help of the project, 1 student thought the project was similar to an algorithm, and 1 student mentioned that they could make small mistakes at first but later write correct codes. When the opinions in the "In Terms of Time" category were analyzed in detail, 2 students said it did not take much effort, and 1 student said it took some time to write the codes.

> *S20: "It was very nice and fun. Writing code is very good, I liked it very much. It is very fun to make the puppets talk."*
> *S25: "The Dino-Dog project was very fun. I liked it a lot. I did it with a lot of love and fun. I think it was a fun and beautiful project."*
> *S18: "Dino-Dog was very fun and writing code was also very fun. It was very easy to write code. It was very enjoyable to watch the project I made, even if it was short. I did it in a short time because it was easy."*

In the researcher's diary of the Dino-Dog Project lesson, the following statements were made:

> *"The students were very excited again on the way to the laboratory. When I told them about the project, they started to listen to me with curiosity and excitement. While I was explaining the sample application, they were worried that they could do the same process, but when they started doing the project, they realized that there was nothing to worry about. They had a lot of fun*

writing the codes during the lesson. They laughed a lot when they played their last project. The happiness of having accomplished a project was visible in their eyes."

Findings of the Ball Bouncing Project

The data belonging to the checklist applied for the Ball Bouncing Project are given in Table 9.

Table 9. Checklist for the Ball Bouncing Project

Codes	Added by (f)	Did not add (f)
Decorations added 2 scenes	31	1
Baseball added his dummy	23	9
Puppet 1 drew his puppet himself	32	-
Puppet 2 drew his puppet himself	32	-
For the Baseball puppet  added the code block.	32	-
For puppet 2 puppet  added the code block	32	-

When the checklist of the Ball Bouncing Project in Table 9 was examined, it was observed that 31 students in the experimental group added the scene of the props 2, 1 student affected a different scene, 23 students added the Baseball puppet, 9 students added a different puppet, and all students drew Puppet 1 and Puppet 2 correctly. When the code blocks were analyzed, it was found that all students wrote the correct codes for the Baseball and Puppet 2 puppets. Analyzing the data in the Movements Checklist revealed that all students added the codes correctly; however, a few students used a different stage, and especially 9 students used other ball puppets instead of the baseball puppet. Data from the Student Diaries, which were applied to the students after the Ball Bouncing Project lesson, are provided in Table 10.

Table 10.Student diaries related to the Ball Bouncing Project

Category / Code		f
As a means of entertainment		44
	It's a lot of fun	22
	It was very beautiful	9
	I like the game very much	6
	I saw that I could accomplish something	2
	I enjoy writing code very much	2
	It was remarkable	1
	The game was adventurous	1
	I played a lot when the game ended	1
In terms of difficulty		19
	I had some difficulty writing code	10
	Very easy	7
	It was very easy when I got used to writing code	1
	It was not a hassle	1

When Table 10 was examined, it was observed that 44 of the experimental group students' feelings, thoughts, and experiences about the ball-bouncing project were categorized under two headings: "As a Means of Entertainment," while 19 were categorized under "In Terms of Difficulty." Under the category "As a Means of Entertainment," the codes were grouped into 8 subheadings, and under the category "In Terms of Difficulty," the codes were grouped into 4 subheadings. When the opinions in the "As a Means of Entertainment" category were examined in detail, 22 students expressed that it was very entertaining, 9 students said it was very beautiful, 6 students liked the game very much, 2 students felt they could achieve something, 2 students enjoyed writing code very much, 1 student found it remarkable, 1 student described it as adventurous, and 1 student mentioned they played a lot when they finished the game. When the opinions in the "In Terms of Difficulty" category were analyzed in detail, 10 students mentioned having some difficulty in writing code, 7 students said it was very easy, 1 student said it became easy once they got used to writing code, and 1 student said it was not challenging.

> *S11: "Ball bouncing was very nice. I had some difficulties while making it at home, but I played with it a lot after the game was finished. I had a lot of fun. I am very happy that I learned Scratch."*
> *S31: "I had a lot of fun making the game but it was a bit challenging. Although the Ball Bouncing game was challenging, I think it was still a fun game."*
> *S20: "Ball bouncing was very fun. It was fun, beautiful and easy. I liked it very much. The pleasure of this game is different."*

The researcher's diary for the Ball Bouncing Project lesson includes the following statements:

> *"The students were very excited again on the way to the laboratory. They were very happy when I told them about the project. When I played the project I had prepared before, they watched me with amazement. Then a few students also played. They seemed very enjoyable. When I started writing the codes, they had difficulty at first. They thought they couldn't do it, but when I helped them, they did it easily. We finished the project and they played the game they made during the rest of the lesson. There was laughter, anger, and shouting in the class. They said they enjoyed it very much. They even insisted that we play one more lesson."*

Findings of the Star Catching Project

Data on the checklist applied for the Star Capture Project are given in Table 11. When the checklist of the Star Catching Project in Table 24 was examined, it was observed that all of the students in the experimental group affected the Space scene, the Robot puppet, the Starfish puppet, the duration variable, and the score variable. When the code blocks were analyzed, it was found that all students wrote the correct codes for the Robot and Starfish puppets. Analyzing the data in the Movements Checklist revealed that all students added the puppets and codes correctly. This indicates that the students had a clearer understanding of the Scratch program.

Table 11. Checklist for the Annual Catching Project

Codes	Added by (f)	Did not add (f)
Added the Space scene	32	-
He added the robot puppet	32	-
Starfish added his puppet	32	-
Added score variable	32	-
Added duration variable	32	-
For the robot puppet	32	-
  added the code block		
For the Starfish puppet	32	-
 added the code block		

Table 12. Student diaries related to the Star Catching Project

Category / Code		f
As a means of entertainment		50
	It's fun	16
	I had no difficulty in making it	11
	It was beautiful	11
	I like it a lot	9
	I always want to do it	1
	It wasn't very exciting	1
	It was a bit boring	1
In terms of difficulty		12
	I had some difficulty	8
	The codes were long	3
	At first, it seemed strange, but then I understood	1

Data from the Student Diaries applied to the students after the Star Catching Project lesson are given in Table 12.When Table 12 was examined, it was found that the feelings, thoughts, and experiences of 50 experimental group students regarding the Star Catching project were categorized under two headings: “As a Means of Entertainment” (f=50) and “In Terms of Difficulty” (f=12). Under the “As a Means of Entertainment” category, the codes were grouped into 7 subheadings, while under the “In Terms of Difficulty” category, the codes were grouped into 3 subheadings. Detailed analysis of the opinions in the “As a Means of Entertainment” category revealed that 16 students found the project fun, 11 had no difficulty doing it, 11 thought it was very beautiful, 9 liked it very much, 1 always wanted to do coding, 1 thought the game was not very exciting, and 1 thought it was a little boring. Detailed analysis of the opinions in the “In Terms of Difficulty” category showed that 8 students thought it was a little difficult, 3 thought the codes were long, and 1 found the game strange at first but understood it later.

> *S14: "Although I had a little difficulty, it was very good. I like to play the game after making it. I have a lot of fun while playing, I enjoy doing it very much. I am glad that we make games. I want to do it all the time."*
> *S9: "It is a little difficult to make a game, but it is very fun and very good to play. It is like downloading a game and playing it."*
> *S10: "The project we did today was very good. I had some difficulties while doing it, but the end was very good. We played a perfect game. I thank my teacher very much."*

The researcher's diary of the Star Capture Project lesson also contains the following statements:

> *"The students were very excited again on the way to the laboratory. They were very happy when I told them about the project. They were very happy when I played the project I prepared for them. I explained how to write the codes. This time they found the codes a bit long and started asking if they could do it. They had a hard time at first, but when they understood the logic, they did it easily. After finishing the project, they started to play the game they made as in other lessons. I think this is the most enjoyable part of the lesson. Both the happiness of accomplishing something and the joy of playing the game they made..."*

Findings of the Labyrinth Project

Data on the checklist applied for the Labyrinth Project are given in Table 13. When the checklist of the Labyrinth Project in Table 13 was examined, it was found that all of the students in the experimental group affected the labyrinth image from the computer, the Giga puppet, the Penguin 2 puppet, and the duration variable. Analysis of the code blocks revealed that all of the students wrote the correct codes for the Giga puppet. Further analysis of the checklist data showed that all of the students added the puppets and codes correctly. This indicates that the students have gained a clearer understanding of the Scratch program.

Data from the Student Diaries applied to the students after the Labyrinth Project lesson are given in Table 14. When Table 14 was examined, it was found that 44 of the experimental group students' feelings, thoughts, and experiences about the Labyrinth project were categorized under "As a Means of Entertainment," while 19 students' opinions were categorized under "In Terms of Difficulty." Under the "As a Means of Entertainment" category, the codes were grouped into 7 subheadings, and under the "In Terms of Difficulty" category, they were grouped into 4 subheadings.

Table 13. Checklist for the Labyrinth Project

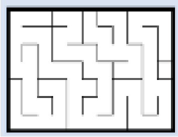
Codes	Added by (f)	Did not add (f)
 added the image from the computer	32	-
Giga added his dummy	32	-
Penguin 2 has added his puppet	32	-
Added duration variable	32	-
For the Giga puppet   added the code block.	32	-

Table 14. Student diaries related to the Labyrinth Project

Category / Code		f
As a means of entertainment		44
	It was very beautiful	21
	It was a lot of fun	13
	It's fun to play the games I make	2
	One of my favorite projects	2
	Remarkable	2
	I wish Scratch would never end	2
	Beyond perfect	2
In terms of difficulty		19
	It was a little hard to play	9
	I had some difficulty	8
	Pro player level	1
	I had no difficulty in making it, it was easy	1

Upon analyzing the opinions in the "As a Means of Entertainment" category in detail, 21 students expressed that it was very beautiful, 13 found it very entertaining, 2 enjoyed playing the games they created, 2 considered it one of their favorite projects, 2 found it remarkable, 1 did not want the Scratch lesson to end, and 1 considered it beyond perfect. When the opinions in the "In Terms of Difficulty" category were examined in detail, 9 students reported that playing the game was a bit difficult, 8 experienced some difficulty in writing the codes, 1 stated that the game was at a pro-player level, and 1 mentioned that they did not have difficulty in making the game and found it easy.

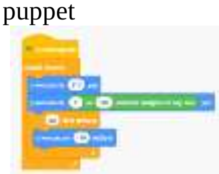
S3: "The game was fun. It was very nice. I had a lot of fun. I think it was a good project. I didn't have any difficulty doing it, I think it was easy."
S30: "It was very nice to make the maze game. Giga and Penguin were very cute. I had a little difficulty catching Giga in the maze, but the game was very nice. I am glad my teacher made us do it."
S20: "The maze project is one of my favorite projects. It is a nice project. But I had a little bit of difficulty when I played it because it went back to the beginning when it said black lines. Still, it's a nice project, it's fun."

In the researcher's diary of the Labyrinth Project lesson, the following statements were made:

"They are very happy when we go to the laboratory every week. They were very surprised when I told them about the project. They started to say how we were going to do it. When they played the project for the first time, they constantly burned out and got very angry. They listened attentively to the project and immediately grasped how to do it. Now it doesn't take time to make a game like before. The children have now grasped the logic. They also got to know the program well. We finished the project together and it was time to play with the games we made. Since this game requires high precision, they constantly hit the edges of the maze and went back to the beginning. Although this made them a little angry, they told us how much they liked the game when they returned to the classroom."

Findings of the Flying Cat Project

Table 15. Checklist for the Flying Cat Project

Codes	Added by (f)	Did not add (f)
Added the Blue Sky scene	32	-
Cat Flying added his puppet	32	-
Added Buildings dummy	32	-
Dragon added his puppet	32	-
Donut added his puppet	32	-
Added the Donut variable	32	-
Added the variable Can	32	-
For the Cat Flying puppet   added the code block.	32	-
For Buildings puppet  added the code block	32	-
For the Dragon puppet   added the code block	32	-
For the donut puppet  added the code block	32	-

The data belonging to the checklist applied for the Flying Cat Project are given in Table 15. When examining the checklist of the Flying Cat Project in Table 15, it was observed that all of the students in the experimental group added the Blue Sky scene, Cat Flying puppet, Buildings puppet, Giga puppet, Dragon puppet, Donut puppet, Donut variable, and Can variable. Upon analyzing the code blocks, it was found that all of the students correctly wrote the codes for the Cat Flying, Buildings, Dragon, and Donut puppets. Furthermore, when the data in the checklist were examined, it was determined that all students added the puppets and codes correctly. This indicates that there are no longer any aspects of the Scratch program that students are unable to understand and that the program has been sufficiently comprehended. Data from the Student Diaries applied to the students after the Flying Cat Project lesson are given in Table 16.

Table 16. Student diaries related to the Flying Cat Project

Category / Code		f
As a means of entertainment		43
	It's so beautiful	19
	It's a lot of fun	14
	I played a lot	4
	Scratch lessons don't end	2
	I like it a lot	1
	One of my favorite projects	1
	It was like real games	1
	The movement of the buildings was spectacular	1
In terms of difficulty		13
	I had no difficulty in making it	8
	I had some difficulty	4
	The codes are a bit long	1

When Table 16 was examined, it was observed that 43 of the experimental group students' feelings, thoughts, and experiences about the Flying Cat project were categorized under "As a Means of Entertainment," while 13 were categorized under "In Terms of Difficulty." Under the category "As a Means of Entertainment," the codes were grouped into 8 subheadings, and under the category "In Terms of Difficulty," the codes were grouped into 3 subheadings. Upon a detailed analysis of the opinions in the "As a Means of Entertainment" category, 19 students expressed that it is very beautiful, 14 found it very entertaining, 4 played a lot with the game they created, 2 did not want the Scratch lessons to end, 1 liked the project very much, 1 considered it one of their favorite projects, 1 felt it was like real games, and 1 found the movement of the buildings eye-catching. When the opinions in the "In Terms of Difficulty" category were examined, 8 students reported having no difficulty in making the project, 4 experienced some difficulty, and 1 felt that the codes were a bit long.

> *S32: "The flying cat game was like real games. It was one of the most beautiful things. I played a lot. The dragon moves and I always miss it. It was as fun to make as it was to play."*
> *S2: "It is a very nice project. The codes are a bit long but it is very fun. I made the game and played it for an hour."*
> *S9: "The flying cat was very beautiful. The movement of the buildings in the game was eye-catching. The cat seemed to fly. It was fun to catch the spawns and escape from the dragon. I am glad my teacher made us do it."*

The researcher's diary of the Flying Cat Project lesson includes the following statements:

> *"They were very happy when we went to the laboratory this week. They were very curious about the project. The name of the airplane cat excited them. They had a lot of fun playing the project I did. They said it was very easy to play this project. I don't know, maybe they got used to Scratch... The codes were a bit long, but they didn't have any difficulty in writing them. Now they are doing projects in a shorter time. This makes me so happy. I am proud to see my students using the Scratch program. After finishing the project today, we had more time to play the game. The children played the game until the end of the lesson. They told each other the scores they got from the game. The lesson was very enjoyable. Children leave the Scratch lessons very happy and having fun."*

Findings of the Color Capture Project

Data on the checklist applied for the Color Capture Project are given in Table 17.

Table 17. Checklist for the Color Capture Project

Codes	Added by (f)	Did not add (f)
Added Blue Sky 2 scene	32	-
Ball added his puppet	32	-
 Color puppet uploaded from computer	32	-
Added score variable	32	-
For Ball puppet  added the code block.	32	-
For color puppet  added the code block	32	-

When the checklist of the Color Capture Project in Table 17 was examined, it was found that all of the students in the experimental group added the Blue Sky 2 scene, and the Ball puppet, uploaded the Color puppet from the computer and added the Score variable. Upon analyzing the code blocks, it was observed that all of the students correctly wrote the codes for the Ball and Color puppets. A further examination of the data in the checklist revealed that all of the students correctly added the puppets and codes. This indicates that the Scratch program is now more clearly understood by the students. The data of the Student Diaries applied to the students after the Color Capture Project lesson are given in Table 18.

Table 18. Student diaries related to the Labyrinth Project

Category / Code		f
As a means of entertainment		43
	Beautiful	15
	It's a lot of fun	14
	I like it a lot	5
	I didn't like it very much	2
	It was boring	2
	Remarkable	1
	Like a brain teaser	1
	These codes are important for my future games	1
	My favorite project	1
	Like a real game	1
In terms of difficulty		13
	The codes were easy to write	6
	It was a bit difficult	4
	I had a hard time downloading the color puppet	3

When Table 18 was examined, it was found that the feelings, thoughts, and experiences of 43 experimental group students regarding the Color Capture project were gathered under two categories: “As a Means of Entertainment” and “In Terms of Difficulty,” with 13 opinions falling under the latter. Under the category "As a Fun Tool," the codes were grouped under 10 subheadings, while under the category "In Terms of Difficulty," the codes were grouped under 3 subheadings. Upon examining the opinions in the “As a Means of Entertainment” category in detail, 15 students found it beautiful, 14 thought it was very entertaining, 5 liked the project very much, 2 did not like it very much, 1 found it boring, 1 found it remarkable, 1 likened it to an intelligence game, 1 believed the codes were important for future games, 1 considered it their favorite project, and 1 thought it was like a real game. In the “In Terms of Difficulty” category, 6 students thought it was easy to write the codes, 4 found it a little difficult, and 3 had difficulty in downloading the color puppet.

> *S25: "It took a very short time, but I had a hard time loading it. I liked it very much. I like projects like this very much. I always want to do it."*
> *S2: "I had a hard time downloading the color puppet and I didn't like the game very much, but the game was still good. I think these codes are important for the games I will make in the future."*
> *S9: "I think it was very fun. I enjoyed doing it and I liked it very much. I would like to make games similar to the color capture game. It was very easy and I have fun doing it."*

The following statements are also included in the researcher's diary of the Color Capture Project lesson:

> *"When they first heard about the color capture project, they didn't make much sense. When they started playing the game I made, they generally enjoyed it. However, I also had bored students. They said that the ball landed too fast and that they could not change colors immediately and that they were bored because of this. I noticed that they had fun when they started to write the codes themselves. There was a problem in the beginning about downloading the color puppet from the computer, but then they easily solved it. When they finished the project and started to play, they were very happy again. There was the happiness of doing another project. When they left the laboratory, they said they had a lot of fun again."*

Table 19. Checklist for the Wheel of Fortune Project

Codes	Added by (f)	Did not add (f)
Decorations added 1 scene	32	-
 Added the Wheel of Fortune puppet	32	-
Puppet 1 drew his puppet himself	32	-
Giga added his dummy	32	-
Added score variable	32	-
For the passionflower puppet  added the code block.	32	-
For the Giga puppet  added the code block.	32	-
For Puppet 1 puppet  added the code block.	32	-

Findings Related to the Wheel of Fortune Project

Data on the checklist applied for the Wheel of Fortune Project are given in Table 19. When the checklist of the Wheel of Fortune Project in Table 19 was examined, it was observed that all the students in the experimental group added the Props 1 scene, the Wheel of Fortune puppet, the Giga puppet, the score variable, and independently drew the Puppet 1 puppet. An analysis of the code blocks revealed that all students correctly coded the Wheel of Fortune, Giga, and Puppet 1 puppets. The data in the checklist confirmed that all students accurately added the required puppets and codes. This indicates that the students have developed a clear understanding of the Scratch program. Data on the Student Diaries applied to students after the Wheel of Fortune Project lesson are given in Table 20.

Table 20. Student diaries on the Wheel of Fortune Project

Category / Code		f
As a means of entertainment		35
	It was very beautiful	20
	It's a lot of fun	10
	It was the best game so far	4
	It was boring to play	1
In terms of difficulty		13
	The codes were easy	7
	A little bit tricky	4
	The codes were long	2

When Table 20 was examined, it was observed that 35 of the experimental group students' opinions about the Wheel of Fortune project were categorized under "As a Means of Entertainment," while 13 were categorized under "In Terms of Difficulty." In the "As a Means of Entertainment" category, the codes were grouped into 4 subheadings, and in the "In Terms of Difficulty" category, the codes were grouped into 3 subheadings. Detailed analysis of the “As a Means of Entertainment” category revealed that 20 students found the project very beautiful, 10 found it very entertaining, 4 considered it the best game they had created so far, and 1 found playing the game boring. In the “In Terms of Difficulty” category, 7 students found it easy to write the codes, 4 felt that creating the game was somewhat challenging, and 2 thought the codes were lengthy.

> *S27: "I think it was very good. I had a lot of fun doing it. I always want to do it. I hope all my projects will be complete. Because I had a lot of fun. It is very fun to make games similar to the Wheel of Fortune. I think it was very easy."*
> *S18: "Wheel of Fortune was a very nice project. The more I turned it, the more points I got. Wheel of Fortune was colorful. I got 1000 only once and I was very happy."*
> *S13: "The codes were long. It was fun and beautiful. It was easy and I didn't have any difficulty in doing it. I liked the game. It was a bit long but it was still good."*

The researcher's diary of the Wheel of Fortune Project lesson includes the following statements:

> *"The Wheel of Fortune project was one of the projects that attracted the attention of the students. They said that they saw this game a lot on the internet. But they expressed that they did not know how to do it. They were very happy when they started writing the codes. They had no difficulty in writing codes this week. Only because the codes were a bit long, did they start to ask when it would be over. I think they were a bit bored... When the game was over, they started playing again. They didn't like the fact that luck was very important in this game. Sometimes they were bankrupt all the time. But they still seemed to have a lot of fun and were happy. The process went very well. Now we are coming to the end of our application. The children don't want the Scratch lessons to end..."*

Findings of the Jumping from Obstacle Project

The data belonging to the checklist applied for the Jumping from the Barrier Project are given in Table 21. When examining the checklist of the Jumping Through Obstacles Project in Table 21, it was observed that all students in the experimental group included the Boardwalk scene, the Pico Walking puppet, the Tree 1 puppet, and the following Block puppets: Block-G, Block-A, Block-M, Block-M, Block-E, Block-O, Block-V, Block-E, Block-R, as well as the Score variable.

Table 21. Checklist for the Jumping the Hurdle Project

Codes	Added by (f)	Did not add (f)
Boardwalk scene added	32	-
Pico Walking added his puppet	32	-
Tree 1 added his dummy	32	-
Block-G added his dummy	32	-
Block A added his dummy	32	-
Block M added his dummy	32	-
Block E added his dummy	32	-
Block-O added his dummy	32	-
Block V added his dummy	32	-
Block E added his dummy	32	-
Block-R added his dummy	32	-
Added score variable	32	-
For the Pico Walking puppet  added the code block.	32	-
For Tree 1 puppet  added the code block	32	-
For Block-G puppet  added the code block	32	-
For Block-A puppet  added the code block	32	-
For Block-M puppet  added the code block	32	-
For Block-E puppet  added the code block	32	-
For Block-O puppet	32	-

 added the code block		
For Block-V puppet	32	-
 added the code block		
For Block-E puppet	32	-
 added the code block		
For Block-R puppet	32	-
 added the code block		

Analysis of the code blocks revealed that all students correctly wrote the codes for Pico Walking, Tree 1, and all Block puppets. The checklist data indicated that all students successfully added the required puppets and codes. The detailed selection and completion of this project demonstrated the students' advanced understanding of the program. The inclusion of all puppets and accurately written codes confirms that the Scratch program has been comprehensively understood by the students. The data of the Student Diaries applied to the students after the Jumping through Obstacles lesson are given in Table 22.

Table 22. Student diaries related to jumping over obstacles

Category / Code		f
As a means of entertainment		49
	It was a lot of fun	26
	It was very beautiful	13
	The game was adventurous	4
	I was very excited when I played	2
	It was remarkable	1
	I enjoyed doing it	1
	It was like real games	1
	It was my best project	1
In terms of difficulty		19
	The codes were too long	16
	I got tired of writing the code	2
	I had no difficulty writing the code	1

When Table 22 was examined, it was found that the opinions of 49 experimental group students about the Jumping Through Obstacles project were grouped under two categories: “As a Means of Entertainment” and “In Terms of Difficulty.” Under the “As a Means of Entertainment” category, 26 students described the project as very entertaining, 13 found it very beautiful, 4 thought it was adventurous, 2 felt very excited while playing, and 1 student each remarked that it was remarkable, enjoyable to create, similar to real games, and the best project they had ever done. Under the “In Terms of Difficulty” category, 16 students expressed that the game's codes were too long, 2 felt tired while writing the codes, and 1 reported no difficulty in writing them. Overall, while the project was widely regarded as engaging and enjoyable, the length of the codes posed a challenge for some students, suggesting room for simplifying or better pacing the exercises.

S31: "The game was very good. It was very enjoyable to play, but the codes were very long. I got tired while writing the codes but it was worth it."

S10: "I did not have any difficulty in making the game. It was like real games. I enjoyed it very much."
S32: "I was very excited while playing the game because it is necessary not to touch the obstacles while jumping over them. It was the best project I have ever done. I am glad that our teacher taught us this game."

The following statements are also included in the researcher's diary of the Jumping from the Barrier Project lesson:

"The children were both very happy and a little sad this week. Because this week we are doing the last Scratch lesson. They wished it would not end. We finished our application with a project worthy of the last week. The children were very surprised when they saw the codes because the codes were long. They started to say that we couldn't do it, but when they started writing the codes, they realized that the codes were not difficult even though they were long. Now they have a good grasp of the codes. They can even do long codes. This is a great happiness for me because I cannot tell you how happy it made me to have given a training that I have been thinking about for years and most importantly to see that it was learned by the students. When the children finished the project, they started to play. Playing with the game was more fun than writing the codes. They played with great pleasure. At the end of the lesson, they left very happy again, just like the other weeks. We have completed our last week without any problems and in a pleasant way."

Findings Related to Student Opinions

The students in the experimental group were asked, "What kind of program do you think Scratch is?" The categories, codes, and their frequency values, based on the students' responses, are presented in Table 23.

Table 23. Ideas about the Scratch Program

Category / Code		f
Emotionally		36
	It's a lot of fun	12
	It was very beautiful	8
	Magnificent	7
	Super for making games	4
	Coding is exciting	3
	It's great.	2
Cognitive aspects		9
	A program that teaches codes	4
	A program that improves our intelligence	2
	A program that teaches algorithms	2
	People think they are informaticians	1

When the feelings, thoughts, and experiences of the experimental group students about the Scratch program were analyzed as shown in Table 23, it was found that 36 opinions were categorized as "Emotional" and 9 as "Cognitive." Under the "Emotional" category, the codes were grouped into six subheadings, while under the "Cognitive" category, the codes were grouped into four subheadings. In the "Emotional Perspective" category, 12 students stated that Scratch is very fun, 8 described it as very beautiful, 7 found it magnificent, 4 expressed that making games is super, 3 mentioned that coding is exciting, and 2 referred to it as great. In the "Cognitive Perspective" category, 4 students remarked that Scratch is a program that teaches coding, 2 highlighted it as a program that develops intelligence, 2 considered it as a program that teaches algorithms, and 1 student mentioned that using Scratch makes them feel like an informatician while writing code.

S8: "It is a very fun program. I enjoyed doing it very much."
S5: "It was very nice and coding excites you. While coding, you think you are a cognitive scientist."
S15: "Scratch is a program that teaches codes. It is both very fun and very enjoyable."

The experimental group students were asked the question, "Did you like the Scratch program? Why?" The categories, codes, and their frequency values derived from the students' responses are presented in Table 24.

Table 24. Liking the Scratch Program

Category / Code		f
Yes		38
	It's fun	19
	I learned to code	7
	We made a game	6
	It was easy	2
	My dream has come true	1
	I felt like a scientist while making a game	1
	I was proud of myself when I learned about the program	1
	It develops our imagination	1

When Table 24 was examined, it was observed that 38 student opinions fell under the “Yes” category regarding their liking for the Scratch program, reflecting their feelings, thoughts, and experiences. The codes within the “Yes” category were organized into eight subheadings. A detailed analysis of the "Yes" responses revealed that 19 students found the program fun, 7 students appreciated learning coding, 6 students enjoyed creating games, 2 students found the program easy, and 1 student expressed that the program made their dream of learning coding come true. Additionally, 1 student mentioned feeling like a scientist while creating games, another student felt proud upon mastering the program, and 1 student believed that the program helped develop their imagination.

> *S20: "I liked the Scratch program because it contains algorithms. I had no difficulty in writing the codes. I participated because it was fun. I was proud of myself when I wrote all the codes correctly. At the same time, when I knew and wrote the codes, it was as if I entered the digital world."*
> *S17: "Yes, I liked it because we can make games. We do coding. It is fun and we learn how to make games."*
> *S30: "I liked the Scratch program very much because I felt like a scientist while making the games. I was in the air, in short, making games was really fun."*

The students in the experimental group were asked the question, "How did you feel while making games with Scratch?". The categories/codes and their frequency values obtained from the students' responses are provided in Table 25.

Table 25. Feelings while making games with Scratch

Category / Code		f
Emotionally		45
	I enjoyed it very much	21
	I'm so happy	16
	I'm so excited	5
	I was scared at the beginning but it didn't happen as I feared	2
	I'm not bored	1
It was the best time of my life		1
I relax while coding.		1
I have a new hobby		1

When Table 25 was examined, it was seen that 45 of the experimental group students' opinions were grouped into 4 categories: "Emotionally," 1 as "The Best Time of My Life," 1 as "I Relax While Writing Code," and 1 as "Acquiring a New Hobby." Under the "Emotionally" category, the codes were grouped into 5 subheadings. When the opinions in the "Emotionally" category are examined in detail, 21 of the student's opinions were that they enjoyed it very much, 16 were very happy, 5 were very excited, 2 were afraid at the beginning but it was not as they feared, and 1 was not bored.

> *S32: "I had fun, I enjoyed it, I felt warm inside as if the characters were real while doing the project. I felt very good, I was relaxing while writing code."*
> *S12: "I was excited when I was making a game with Scratch because I had questions in my mind such as can I succeed or did you write the codes correctly? At the end of the game, if I succeeded, if the game was correct, I was very happy."*
> *S5: "I felt very excited when I made my first game with Scratch. I was very happy when I played my first game. It was one of the happiest moments of my life.”*

When the experimental group students were asked, "Which project did you enjoy doing the most? Why?", the categories/codes and their frequency values obtained as a result of the answers received from the students are provided in Table 26.

Table 26. The most enjoyable projects

Category / Code		f
Ball Bouncing		14
	It was as much fun as real games	7
	It was beautiful	5
	It was easy	1
	Because it's my own game	1
I like them all		11
	They were all beautiful	4
	I enjoyed doing it all	3
	They were all fun	2
	Educational	1
	Improving our intelligence	1
Wheel of Fortune		10
	It was a lot of fun	6
	It was colorful	1
	We made the game that everyone loved	1
	It was like the games I played on the internet	1
	Because I wrote the code myself	1
Dino Dog		8
	The characters were funny	4
	Dialogues were good	2
	It was fun	2
Labyrinth		6
	It was fun	3
	The voice in the game was delightful	2
	It was very beautiful	2
Star Chasing		2
	It was very enjoyable	2
Color Capture		2
	It was a lot of fun	1
	It was exciting	1

When the feelings, thoughts, and experiences of the experimental group students regarding the projects they enjoyed were examined, it was found that 14 of the students' opinions were gathered under 7 categories, including "Ball Bouncing Project," 11 under "I liked it all," 10 under "Wheel of Fortune Project," 8 under "Dino Foam Project," 6 under "Labyrinth Project," 2 under "Star Chasing Project," and 2 under "Color Catching Project." Under the "Ball Bouncing Project" category, the codes were grouped under 4 subheadings. When examined in detail, 7 of the students found it as fun as real games, 5 thought it was beautiful, 1 found it easy, and 1 appreciated having their own game. Under the "I Like It All" category, the codes were grouped under 5 subheadings. Detailed analysis showed that 4 students felt all the projects were beautiful, 3 enjoyed doing all of them, 2 found all of them fun, 1 thought they were educational, and 1 felt they enhanced their intelligence. In the "Wheel of Fortune Project" category, the codes were grouped under 5 subheadings. Detailed analysis revealed that 6 students found it very entertaining, 1 thought it was colorful, 1 made a game that everyone loved, 1 felt it resembled internet games, and 1 wrote the code themselves. Under the "Dino-Dog Project" category, the codes were grouped under 3 subheadings. 4 students found the characters funny, 2 appreciated the dialogues, and 2 enjoyed the project overall. In the "Labyrinth Project" category, the codes were grouped under 3 subheadings. Detailed analysis revealed that 3 students found it fun, 2 enjoyed the sound of the game, and 2 thought it was beautiful. Under the "Star Chasing Project" category, the codes were grouped under 1 subheading. 2 students found it very enjoyable. In the "Color Catching Project" category, the codes were grouped under 2 subheadings. 1 student found it very fun, and 1 found it exciting.

> *S19: "I enjoyed the Ball Bouncing game the most because the ball was bouncing and we were trying not to drop it and lose it. It was as fun as real games."*
> *S8: "I enjoyed doing all the games and they were fun. The reason is that it is educational and improves your intelligence."*

S21: "Wheel of Fortune because it is enjoyable like the games on the internet and we make the game that everyone loves. It is very enjoyable"

It seems like you're referring to Table 27, but I don't have the specific details from that table. If you'd like, you can share the data or key points from Table 27, and I can help you structure or analyze the information.

Table 27. Difficulties in creating projects with Scratch

Category / Code		f
I had some difficulty.		14
	I struggled with some projects	4
	When codes are long	3
	Locating code blocks is a bit tricky	3
	When I load something from the computer	2
	Hard to generate code	2
No difficulty at all		10
	It was easy	5
	I enjoyed it very much	2
	My teacher's guidance was good	1
	It was fun to play	1
	It was fun	1
Difficulty		8
	In my first projects	4
	The codes were sometimes confusing	2
	I was having a hard time downloading it from my computer	1
	There was a lot of work to be done	1

When the feelings, thoughts, and experiences of the experimental group students regarding the difficulties they faced in creating projects with Scratch were analyzed in Table 27, it was found that 14 of the students' opinions were grouped under the category "I had some difficulty," 10 of them under "I had no difficulty," and 8 of them under "I had difficulty." The codes under the "I had some difficulty" category were grouped under 5 subheadings, the codes under the "I had no difficulty" category were grouped under 5 subheadings, and the codes under the "I had difficulty" category were grouped under 4 subheadings. In the "I had some difficulty" category, 4 students reported difficulty with certain projects, 3 mentioned difficulty when the codes were long, 3 found it challenging to locate code blocks, 2 struggled when loading files from the computer, and 2 found it hard to create codes. In the "I had no difficulty" category, 5 students felt that the projects were easy, 2 enjoyed them greatly, 1 appreciated the teacher's guidance, 1 found making games enjoyable, and 1 found it fun. In the "I had difficulty" category, 4 students reported difficulty with their first projects, 2 felt the codes were sometimes complicated, 1 had difficulty downloading files from the computer, and 1 found the tasks had too many operations to complete.

S15: "I did not have any difficulty in making a game with Scratch because the codes and characters were easy. Since I did the codes carefully, I did not make any mistakes."
S24: "Yes, I had difficulty in some projects, but in general it was easy. Even though it was difficult, I liked it very much."
S31: "Yes, it was difficult because choosing the characters, making their codes one by one, downloading the background from the computer and downloading it to Scratch... there is a lot to do."

Conclusion and Discussion

According to the results of the Algorithmic Thinking Skill Scale, which was developed to examine the effect of the Scratch program on coding education for 4th-grade primary school students, a significant difference was found between the post-test and pre-test scores of the experimental group, with post-test scores being higher. This indicates that the Scratch program had a significant impact on the algorithmic thinking skills of primary school students. When reviewing the literature, it was noted that few studies have specifically addressed the effect of the Scratch program on algorithmic thinking skills. For example, Bahar (2023) found that Scratch applications had a significant impact on algorithmic thinking, critical thinking, and creativity, but did not significantly affect problem-solving skills. Similarly, Cakıcı and Ozdemir (2022) concluded that coding education without computers led to a statistically significant improvement in students' problem-solving and algorithmic thinking skills. The findings from these studies are consistent with the results of this study, supporting the idea that Scratch programming positively influences algorithmic thinking skills.

According to the results of the checklists for the Scratch lessons and the projects developed, it was concluded that some of the students in the experimental group initially misplaced certain puppets and decorations in their first projects. However, as they became more familiar with the program, all students were able to complete the projects. This suggests that with practice and experience, the students' understanding and skills in using Scratch improved, enabling them to finish the tasks accurately.

According to the results obtained from the student diaries regarding the Scratch lessons and the projects developed, the opinions of the experimental group students were generally positive. They expressed that they found the projects fun, and they liked the projects very much. The students reported that they enjoyed writing code and did not experience difficulty in completing the projects. Additionally, they mentioned that they would like to continue with Scratch lessons. The students also stated that playing the games they created was highly enjoyable, as it gave them a sense of accomplishment. They found the projects remarkable and expressed excitement while playing the games they developed.

According to the results obtained from the students' opinions about the Scratch program, the experimental group of students expressed that the Scratch program was very fun, visually beautiful, and wonderful. They stated that making games was an exciting and enjoyable experience. The students mentioned feeling very happy and excited while working with Scratch, although some initially felt scared, they found it less challenging than they anticipated. They enjoyed the process very much and expressed that they liked all the projects. Additionally, the majority of the students reported not facing difficulty in creating the projects. When reviewing the literature, it is noted that there are limited studies on primary school students' opinions about the Scratch program. For instance, Yurtbakan (2022) found that gifted students viewed the Scratch program as fun during the qualitative phase of their study. Ozturk (2021) reported that middle school students found games designed with Scratch interesting, visually appealing, and entertaining. Similarly, Cubukluoz (2019) concluded that students found games created using Scratch to be interesting and fun. The findings from these studies are consistent with the results of this study.

Recommendations

In this study, where the teaching of the Scratch program was conducted, it was observed that elementary school students easily learned to use the Scratch program. Therefore, coding education can be provided to elementary school students, and the Scratch program can be taught. The Scratch program can be taught over a longer period, giving students more opportunities to create games. Students can be encouraged to make different games using the Scratch program. The Scratch program can also be taught to 3rd-grade elementary school students, and teachers can use Scratch activities to increase students' interest and motivation in the lesson. The effects of coding education on other cognitive skills can be examined, and due to the lack of sufficient studies on coding education in elementary schools in the literature, researchers can focus on this area. Additionally, the Ministry of National Education can include coding education in the elementary school curriculum.

Scientific Ethics Declaration

*The authors declare that the scientific ethical and legal responsibility of this article published in JESEH journal belongs to the authors.

*In this study, ethical approval was obtained from the Social and Human Sciences Research Ethics Committee of Inonu University with the approval number 2022/14-8, dated 30.06.2022.

Acknowledgements or Notes

This study is based on the doctoral thesis titled "An Examination of Coding Education Practices in terms of Elementary School Students' Algorithmic Thinking Skills and Student Opinions," completed by the first author under the supervision of the second author (Doctoral Thesis, İnönü University, Malatya, Turkey, 2024).

References

Akpınar, Y., & Altun, A. (2014). Bilgi toplumu okullarında programlama egitimi gereksinimi. *Ilkogretim Online Dergisi, 13*(1), 1-4.

Arı, E. (2016). *Orneklerle algoritma ve C# programlama*. Seckin Yayıncılık.

Aytekin, A., Cakır, F. S., Yucel, Y. B. &Kulaozu, I. (2018). Algoritmaların hayatımızdaki yeri ve onemi. *Avrasya Sosyal ve Ekonomi Arasstırmaları Dergisi, 5*(7), 143-150.

Bahar, N. (2021). *Scratch'ın cocukların ingilizce ve bilissel beceri uzerine etkisi*. (Master's thesis, Middle East Technical University).

Brown, W. (2015). *Introduction to algorithmic thinking*. Retrieved from www.cs4fn.com/algoritmicthinking.php.

Buyukozturk, S., Kılıc- Cakmak, E., Akgun, O. A., Karadeniz, S., & Demirel, F. (2020). *Bilimsel arastırma yontemleri* (28th ed.). Ankara: Pegem Akademi.

Creswell, J. W. (2017). *Arastırma deseni nitel, nicel ve karma yontem yaklasımları*. Egiten Kitap.

Cakıcı, Y., & Ozdemir, S. M. (2022). Bilgisayarsiz kodlama egitiminin ilkokul ogrencilerinin dikkatini toplama, problem cozme ve algoritmik dusunnme becerileri uzerine etkisi. *Uluslararası Bilim ve Egitim Dergisi*, *5*(3), 235-254.

Calıskan, E. (2020). Code.org etkinliklerinin ortaokul ogrencilerinin problem cozme becerilerine ve programlama oz-yeterliklerine etkisinin incelenmesi. *Journal of Instructional Technologies and Teacher Education, 9*(2), 114-124.

Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., & Woollard, J. (2015). *Computational thinking-A guide for teachers.* Retrieved from https://community.computingatschool.org

Cubukluoz, O. (2019). *6. sınıf öğrencilerinin matematik dersindeki ogrenme zorluklarının Scratch programıyla tasarlanan matematiksel oyunlarla giderilmesi: Bir eylem araştırması* (Master's thesis, Bartın University).

Delice, A. (2018). Karma yöntem desen seçimi. In *Karma yontem arastırmaları tasarımı* (pp. 61-114). Anı Yayıncılık.

Demirer, V., & Sak, N. (2016). Programming education and new approaches around the world and in Turkey. *Egitimde Kuram ve Uygulama, 12*(3), 521-546.

Demirkol, Z. (2017). *Ogretmen ve ebeveyn rehberliginde çocuklar icin kodlama.* Pusula 20 Teknoloji ve Yayıncılık.

Dickins, R., Melmoth, J., & Stowell, L. (2016). *Scratch ile yeni baslayanlar icin kodlama*. Altın Kitaplar Yayınevi.

EBA. (2024). *Egitim bilisim agı.* Retrieved from https://www.eba.gov.tr/arama?q=kodlama.

Ercil- Cagıltay, N., & Fal, M. (2015). *Scratch ile programlamayı ogreniyorum*. ODTU Yayıncılık.

Erol, O. (2020). *Kodlama ogretimi programlamadan kodlamaya yaklasımlar ve ornek uygulamalar.* Anı Yayıncılık.

Futschek, G. (2006). Algorithmic thinking: The key for understanding computer science. *Proceedings of the 2nd International Conference on Informatics in Secondary Schools: Evolution and Perspectives (ISSEP)* (pp.159–168).

Genc, Z., & Karakus, S. (2011). Tasarımla ogrenme: Egitsel bilgisayar oyunları tasarımında Scratch kullanımı. *5th International Computer & Instructional Technologies Symposium (ICITS)*.

Gezgin, D. M., Ozcan, S. N., Ergun, K., Kose, O., & Emir, N. (2017) Bilgisayar programlama egitiminde Scratch programı kullanımına iliskin lise öğrencilerinin gorusleri. *Uluslararası Bilimsel Araştırmalar Dergisi*, 182-188.

Gibson, J.P. (2012). Teaching graph algorithms to children of all ages. *ITiCSE '12:17th Annual Conference on Innovation and Technology in Computer Science Education*, 34-39.

Gultekin, M., Gundogan- Bayır, O., & Yasar, E. (2020). Karma arastırma yontemi. B. Oral & A. Coban (Ed.), *Kuramdan uygulamaya egitimde bilimsel arastırma yontemleri* (pp. 317-354). Ankara: Pegem.Akademi.

Johnson, R. B., Onwuegbuzie, A. J. & Turner, L. A. (2007). Toward a definition of mixed methods research. *Journal of Mixed Methods Research*, *1*(2), 112-133.

Kızılkaya, M. & Sart, G. (2017). *Super Scratch programlama yolculugu*. Aba Yayın.

Kordaki, M. (2012). Diverse categories of programming learning activities could be performed within Scratch. *Procedia-Social and Behavioral Sciences, 46,* 1162-1166.

Küçükkoç, I. (2020). *Algoritma ve programlamaya giris ders notları*. Retrieved from https://birecik.harran.edu.tr/assets/uploads/other/files/birecik/files/3Algoritma_ve_Programlamaya_Giri%C5%9F_Ders_Notlar%C4%B1.pdf

Lee, Y.J. (2011). Scratch: Multimedia programming environment for young gifted learners. *Gifted Child Today, 34*(2), 26-31.

Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The Scratch programming language and environment. *ACM Transactions on Computing Education (TOCE)*, *10*(4), 1-15.

Mclennan, D.C. (2017). Creating coding stories and games. *Teaching Young Children*, *10*(3), 18-21.

Olsen, F. (2000). Computer scientist says all students should learn to think 'algorithmically. In *The chronicle of high education.* Springer.

Ozturk, A. (2021). *Ortaokul matematik ogretmenlerinin Scratch programıyla tasarladıkları oyunların ogretmen ve ogrenci goruseri dogrultusunda incelenmesi: cebirden yansımalar*(Master's thesis, Bartın University).

Ozdamar, K. (2017). *Egitim, sağlık ve davranıs bilimlerinde olcek ve test gelistirme yapısal esitlik modellemesi IBM SPSS, IBM SPSS, AMOS ve MINITAB uygulamalı*. Nisan Kitabevi.

Pallant, P. (2017). *SPSS kullanma kılavuzu SPSS ile adım adım veri analizi.* (S. Balcı & B. Ahi, Trans.). Anı Yayıncılık.

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

Sayın, Z. & Seferoglu, S. S. (2016). Yeni bir 21. yuzyıl becerisi olarak kodlama eğitimi ve kodlamanın eğitim politikalarına etkisi. *Akademik Bilisim Konferansı*, 3-5.

Scratch. (2023). *Scratch about*.Retrieved From: http://scratch.mit.edu/about/.

Su, G. (2019). *Scratch ile programlama*. KodlabYayın Dagıtım.

Yiğit, M. F. (2016). *Gorsel programlama ortamı ile ogretimin öğrencilerin bilgisayar programlamayı öğrenmesine ve programlamaya karsı tutumlarına etkisinin incelenmesi*. (Master's thesis, On Dokuz Mayıs University).

Yukselturk, E., & Altıok, S. (2015). Bilisim teknolojileri ogretmen adaylarının bilgisayar programlama ogretimine yonelik gorusleri. *Amasya Universitesi Egitim Fakultesi Dergisi*, *4*(1), 50-65.

Yükseltuurk, E., & Ucgul, M. (2018). Blok tabanlı programlama. In Y. Guulbahar & H. Karal (Ed.), *Kuramdan uygulamaya programlama ogretimi* (pp. 273-296). Pegem Akademi.

Yurtbakan, E. (2022). Ozel yetenekli ılkokul ogrencilerinin yazma egilimleri, kaygıları ve tutumlarında Scratch kodlama programının etkisi. *Mersin Universitesi Egitim Fakultesi Dergisi*, *18*(3), 241-258.

Authors Information

Anil Erkan
Toki Firat Primary School
Elazıg/Turkiye
Contact e-mail: *anilgok86@gmail.com*
ORCID iD: https://orcid.org/0000-0002-3036-1208

Sumeyra Akkaya
Inonu University
Malatya/Turkiye
ORCID iD: https://orcid.org/0000-0002-9942-9848