Article History
Received: 03.12.2021
Received in revised form: 13.04.2022
Accepted: 17.05.2022
Article Type: Research Article

International Journal of Contemporary Educational Research (IJCER)

www.ijcer.net

Coding in Preschool Science and Mathematics Teaching: Analysis of Scratch Projects of Undergraduate Students

Didem Karakaya Cırıt[1]
[1]Munzur University, 0000-0002-8606-478X

To cite this article:

Karakaya-Cırıt, D. (2022). Coding in Preschool Science and Mathematics Teaching: Analysis of Scratch Projects of Undergraduate Students. *International Journal of Contemporary Educational Research*, *9*(3), 460-475. https://doi.org/10.33200/ijcer.1031848

International Journal of Contemporary Educational Research
Volume 9, Number 3, September 2022, 460-475 ISSN: 2148-3868

Coding in Preschool Science and Mathematics Teaching: Analysis of Scratch Projects of Undergraduate Students

Didem Karakaya Cırıt[1*]
[1]Munzur University

Abstract

This paper analyzed Scratch projects developed by undergraduate students. The sample consisted of 22 child development students (18 women and four men) in the 2018-2019 academic year. The study adopted an action research design within the scope of a course titled "Teaching Science and Mathematics in Preschool Education." The research was conducted within 14 weeks. In the first four weeks, we provided participants with training on why and how to use Scratch in science and mathematics teaching. In the following ten weeks, participants designed Scratch projects every week based on age groups, topics, and learning outcomes of their choice. Participants evaluated their projects themselves and also received feedback from peers and academics. Each participant designed ten Scratch projects (five for math and five for science). The data consisted of 220 Scratch projects and design logs. The study included a thematic content analysis. In the first weeks, participants knew little about the content of Scratch and used one or two characters and mostly control and look blocks. In the following weeks, they learned more about Scratch and used different blocks.

Keywords: Scratch, Science education, Mathematics education, Preschool education, Coding

Introduction

In recent years, visual programming languages (Alice, code.org, Scratch, etc.) have made coding popular in education. Such programming languages teach young children how to use codes without having to learn complicated code structures (Resnick et al., 2009). They help children create their own stories, animations, interactive games, and simulations to learn math and science skills and topics (Taylor, Harlow & Forret, 2010). Scratch is the most popular block-based software to teach beginners how to code (Zhang & Nouri, 2019). Apps like Scratch are defined as visual programming tools because they are based on the "drag and drop" feature, by which the user can simply choose visual elements and drop them wherever she wants. In other words, they offer a simple interface that allows children to design and produce without making coding errors. Children who receive programming education at an early age are more likely to develop metacognitive skills (divergent thinking, creativity, and channelization) (Atmatzidou & Demetriadis, 2016). Therefore, schools should provide children with coding education from an early age (Calder, 2010). Coding is a 21st-century skill and a part of logical reasoning (European Commission, 2014a). Many European countries have included coding in their curricula to help students develop logical thinking and problem-solving skills. Some countries have integrated coding into their curricula to stimulate employment (Balanskat & Engelhardt, 2015). Scratch teaches young learners how to code and helps them develop higher-order thinking skills (Zhang, Yang, Luan, Yang, & Chua, 2014). It also provides students with the opportunity to learn math and science concepts.

Educational institutions, researchers, teachers, and government agencies such as the NSF have worked to develop sustainable curricula that encourage students to learn programming at an early age. However, some obstacles are encountered in practice. For example, insufficient professional development services for teachers (Buss & Gamboa, 2017); insufficient evidence of the impact of practices using innovative learning environments (for example, Code.org or Scratch) (Kalelioğlu & Gülbahar, 2014; Moreno-León, Robles & Román-González, 2016) It has been suggested that many schools today adopt block-based programming environments such as Scratch to solve their problems. Even though most teachers provided the use of Scratch in

* Corresponding Author: *Didem Karakaya Cırıt*, *ddmkrkycrt@gmail.com*

their classrooms, they were insufficient to evaluate the quality of the projects created by the students and to give feedback to them according to the results (Kwon, Lee & Chung, 2018). Teachers want to answer the question: how to evaluate the quality of students' scratch programs? For example, Dr. Scratch automatically evaluates the scratch program (Moreno-León, Robles, &Román-González, 2015). As another example, students' programming competencies and strategies in Scratch programs include 1-computational concepts (data representation, iteration with certain loops, etc.), 2-computational designs (task sequences, parsing problems, etc.), and 3-it can be evaluated as their computing performance (determination of targets, usability, etc.) (Chao, 2016). Quantitative indicators of scratch programs can show the size and complexity of a program (Aivaloglou & Hermans, 2016).

Literature Review

Using Scratch in Science and Mathematics Education

It is important that students, especially those in the pre-school period, can achieve meaningful and permanent learning in science and mathematics and develop positive attitudes. With the use of Scratch, abstract concepts such as variables in algebra can be learned in mathematics education (Fesakis et al., 2013). In addition to easier learning of abstract concepts containing variables, it also improves students' high-level thinking skills (Monroy Hernandez & Resnick, 2008). Considering that many subjects in science are abstract (e.g., atoms, photosynthesis, global warming, etc.), modeling may be required to make it easier for students to visualize these subjects. Modeling helps us understand its real-world effectiveness by simplifying some of the features of the phenomenon (Weintrop et al. 2016). Students can design and create their own models (Weintrop et al. 2016). Models are important in promoting inquiry, conceptual change, and representative literacy (Brennan & Resnick 2012). The NGSS emphasizes the importance of students creating models (Weintrop et al., 2016). Scratch is also an environment that can be used effectively in model creation.

Scratch

Scratch, which has the most common use as block-based programming, also offers a rich interactive programming opportunity in terms of media. Scratch is a free-to-use graphical programming language (http://scratch.mit.edu/) designed to facilitate and improve technological fluency (Resnick & Silverman, 2005). Scratch is developmentally based on the ideas of Seymour Papert (Papert, 1980) and was developed at the Massachusetts Institute of Technology's (MIT) Media Lab. Papert's ideas support Scratch design with "repairability", as one of its main goals is to assemble, disassemble, and reassemble the building blocks of coding to build what child programmers want (Resnick, 2007). With Scratch, users can solve and reassemble blocks as they logically develop the moves and effects they want. It therefore allows for creative and accessible programming. It has a program interface with three different areas. It is the left part where the programming blocks are located. Each of these blocks has its own function and can be dragged into the script and put together to create programs. With the "drag-and-drop" function in Scratch, the need to remember codes or understand syntax is eliminated (Otrel-Cass, Forret, & Taylor, 2009). Writing a syntactically incorrect script is possible in programming languages. Scripts do not fit together unless they are written from scratch. Therefore, it makes learning and creating programs easier and allows the child programmer to spend more time on the logic and creative elements of the programs (Otrel-Cass, et al., 2009). Using different shapes and combining commands with parts prevents syntactic errors by restricting the structures in the program (Bau, Gray, Kelleher, Sheldon, & Turbak, 2017). The middle part, where the script or program is located, is the area where the programming blocks are combined. In the upper right corner, there is the “scene” where the graphic elements are placed.

Scrach allows users to create their own interactive stories, animations, games, or images. While working individually and collaboratively on interactive stories, games, and animations, 21st century skills such as critical thinking, creativity, communication, and cooperation can be learned, as well as mathematical and computational concepts (Maloney et al., 2010; Resnick et al., 2009). Users can share their created projects online. It can be used for teaching and learning purposes in many subjects, such as mathematics, science, music, and art. Although simple and easy to use, the projects put forward are complex and comprehensive and offer a fun learning environment. While it is enjoyable for children to explore and create creative products with programming, it also contributes to their understanding of embedded programming and mathematical concepts such as location or orientation. Creative problem solving, facilitating logical reasoning, and collaboration are encouraged with Scrach (Peppler & Kafai, 2006). It makes programming accessible and engaging for everyone on Scratch (Resnick et al., 2009). It is easy for those with limited or no programming background to start learning programming and can build more complex programs over time (Sáez-López, Román-González, & Vázquez-Cano, 2016; Su, Yang, Hwang, Huang & Tern, 2014).

Scratch was developed by the Massachusetts Institute of Technology (MIT) to make programming easier. It allows users to create animations, games, stories, and interactive projects and share them online. It runs either online or offline. Its interface consists of five sections: (1) a stage where a Scratch project is physically run; (2) blocks; (3) a script area where blocks are placed; (4) an area to choose a backdrop and sprite, and (5) "go" and "stop" buttons. You can choose different backdrops and program their motion. Each object on the stage is a sprite. You can attach sounds and music to a sprite. Codes are in the form of blocks. You can combine codes to program. For example, we use motion blocks to make our sprite walk or look at blocks to change its color or make it talk.

Evaluation of Scratch

There are different methods and approaches in the evaluation of projects created by students. Three-stage evaluation for visual programming (Wilson, Hainey, & Connolly, 2012); it can be expressed as 1-programming concepts (variables, conditions, etc.), 2-code organization (naming variables, etc.), and 3- interface and usability (authenticity, etc.). In another study, Tseng and Weng (2009) stated that student projects can be done in three ways. These are: 1-Logic (program scenario), 2-Code (program flow, scene, etc.) and 3-resources (sound, graphics, etc.). With the help of a system, automatic evaluation can be carried out using these criteria.

Automatic Assessments

It is a system in which programming tasks are evaluated automatically, does not require any special infrastructure, gives detailed feedback, and can get points in a short time for large groups (Sant, 2009). As an example, "Test My Code (TCM), REACT, and Dr. Scratch" can be given as an example. It has been stated that TCM is useful for use at universities (Vihavainen, Vikberg, Luukkainen, & Pärtel, 2013). REACT was developed by Koh et al. (2014) to assess teachers' computational thinking skills. "Dr. Scratch" (http://www.drscratch.org), which is open to everyone and does not charge any fee for its use, is another automated assessment tool. Moreno-Leon et al. (2015) introduced "Dr. Scratch", a web application that analyzes Scratch programs. Dr. Scratch evaluated computational thinking according to seven criteria: 1-abstraction and problem decomposition; 2-logical thinking; 3-synchronization; 4-parallelism; 5-algorithmic concepts of flow control; 6-user interaction; and 7-data representation. When users present their Scratch programs, Dr. Scratch displays the numeric scores of the criteria (from zero to three), as well as the overall mastery level in terms of basic, development, and mastery. Both students and instructors, Dr. Using Scratch, he can easily evaluate Scratch programs and get instant feedback.

Other Evaluations

In the evaluation of the projects prepared by the students, rating criteria lists, tests, self-evaluation, observation notes, etc. can be used. Another assessment tool is the questions in the Bilge Kunduz activity. It is an international event created to teach students computational thinking. In the activity, students are given 45 minutes to complete 18 questions of varying difficulty. For example, 6 points are given for 6 questions at an easy level, 9 points for 6 questions at a medium level, and 12 points for 6 questions at a hard level. For each wrong question, one third of the correct score of that question is taken and subtracted from the total score according to the difficulty level of that question.

Significance of the Research

The importance of this study can be discussed under four headings when the studies in the literature are examined. First, it allowed preschoolers to create their own science and math Scratch projects for ten weeks based on topics and learning outcomes of their choice. To our knowledge, this is the first study to address Scratch projects within the scope of both science and math. Most research focuses only on one field or does not address any specific field at all. Researchers concentrate on math (Lewis & Shah, 2012); English (Moreno-León & Robles, 2015); information and communication technologies (Yildiz, Cobanoglu & Kisla, 2020); science (Adler & Kim, 2018; Tan, Samsudin, Ismail & Ahmad, 2020); and science concepts (Moreno-León, Román-González, Harteveld & Robles, 2017). The second strength of this study was its scope. Studies examine different components of Scratch education, such as academic performance (Korkmaz, 2018; Tan, Samsudin, Ismail & Ahmad, 2020); computational thinking (Oluk & Korkmaz, 2016; Kwon, Lee & Chung, 2018); pre-programming period (e.g., Java) (Malan & Leitner, 2007; Maloney et al., 2008); and analysis of Scratch projects (Kwon, Lee, & Chung, 2018; Moreno-León et al., 2017; Oluk & Korkmaz, 2016). This study analyzed 220 original Scratch projects. Most researchers analyze Scratch projects that have already been created and archived. The third strength of this study was the way Scratch projects were analyzed. We investigated what types of blocks participants preferred to use and how often they used them, depending on the weeks. Most studies focus on projects created on the Dr. Scratch software program (Adler & Kim, 2018; Altanis & Retalis, 2019; Kwon, Lee & Chung, 2018; Moreno-León et al., 2017; Oluk & Korkmaz, 2016). For example, Moreno-León et al. (2017) compared Scratch projects evaluated by Dr. Scratch (automatically) and experts (manually). Dr. Scratch evaluates the level of development of various aspects of computational thinking. Fifty-three projects created by students were analyzed by both Dr. Scratch and experts. The results show strong relationships between automatic and manual evaluations. The fourth strength of this study was that it focused on the preschool period. There are a handful of studies addressing Scratch use among preschool teachers or preservice preschool teachers. Researchers generally focus on middle school students (Meerbaum-Salant, Armoni & Ben-Ari, 2013; Moreno-León & Robles, 2015; Oluk & Korkmaz, 2016), teachers (Van Zyl, Mentz & Havenga, 2016; Yildiz, Çobanoğlu & Kişla, 2020), and preservice teachers (Altanis & Retalis, 2019; Kwon, Lee & Chung, 2018; Tijani, Callaghan & de Villers, 2020). For example, Moreno-León and Robles (2015) recruited two fourth graders and two fifth graders to conduct an experimental study. The experimental group used Scratch to perform some programming activities on vocabulary and grammar, while the control group adopted a traditional method to address the same topics. The experimental group outperformed the control group. The results showed that coding could be used not only to teach students English but also to help them develop other skills.

Technology has been integrated into the educational process. Therefore, teachers should be able to use their own content to determine students' prior knowledge, to teach new topics, and to evaluate performance. For example, a teacher is expected to create her own content to teach an abstract and hard-to-understand topic, such as the depletion of the ozone layer. She should use Windows Movie Maker to create a video or Scratch to create an animation if she has no access to ready-made content (animation, video, etc.) or if the content she has is not suitable for the learning outcomes she intends to implement. She should be able to develop content based on a predetermined pedagogical approach and have the knowledge to encourage students to develop their own content. Preschool teachers and preservice teachers play a key role in helping young learners develop those skills.

The Aim of the Study

The main purpose of the study is to analyze the code usage frequency and content of Scratch projects prepared by students in science and mathematics subjects. Evaluation of Scratch projects will contribute to students' gaining a perspective of revealing the parts they have difficulty in and designing learning activities in an original way in science and mathematics education. This study analyzed science and math Scratch projects created by undergraduates. The research questions are as follows:

- What is the frequency of use of blocks in Scratch projects on science and mathematics subjects created by students?
  - What are the science and math subjects in Scratch projects created by students? What is the frequency of use of "topics"?
  - What is the frequency of use of "number of characters" in Scratch projects on science and mathematics subjects created by students?
  - What is the frequency of use of "movement blocks" in Scratch projects on science and mathematics subjects created by students?
  - What is the frequency of use of the "view block" in Scratch projects on science and math topics created by students?
  - What is the frequency of use of "control block" in Scratch projects on science and mathematics subjects created by students?
  - What is the frequency of use of "other blocks" in Scratch projects on science and mathematics subjects created by students?
  - What is the frequency of use of "total blocks" in Scratch projects on science and mathematics subjects created by students?

Method

This paper analyzed preschool science and math Scratch projects created by undergraduates. The study adopted an action research design. Action research aims to examine people's problems, especially those that concern society, with conciliatory, democratic, participatory strategies (Berg, 2001). Action research in the field of education, on the other hand, is a systematic process aimed at solving and improving the problems experienced (Tomal, 2010). Action research takes place differently in the literature (Berg, 2001; Bogdan & Biklen, 2007; Hendricks, 2006; Mills, 2003; O'Brien, 2001). Berg (2001) classified three types (technical, practical, and liberating); Mills (2003) classified two types (critical and applied); Bogdan and Biklen (2007) classified two types (political and participatory); and Hendricks (2006) classified four types (collaborative, critical, in-class, and participatory). This type involves a social and collaborative process. Action research tries to solve the problems experienced in practice.

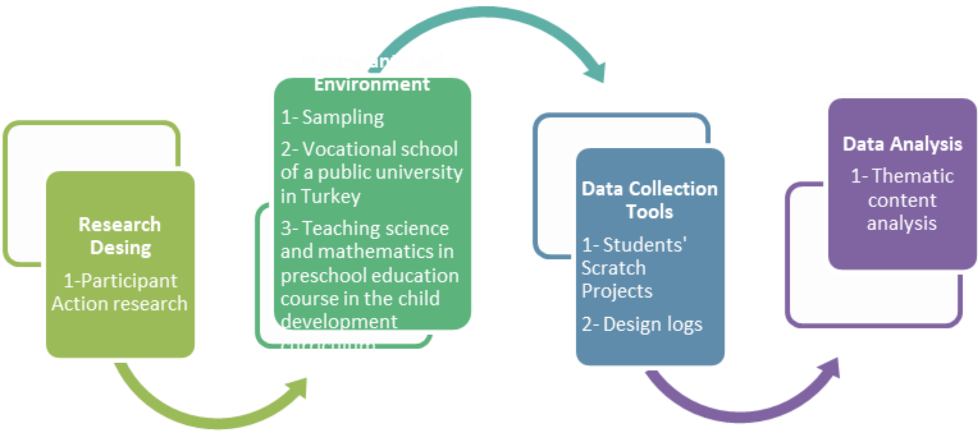


Figure 1. Research Process

Study Group

The sample consisted of 22 fourth-year students (18 women and four men) from the department of child development of a public university in the 2018-2019 academic year. None of the participants had used Scratch before. Most participants had no Internet access or a computer. It was determined that 12 of the students (10 girls and 2 boys) did not have a personal computer and did not have access to the internet. However, it was determined that 10 of the students (8 women and 2 men) had personal computers and only 6 of the women had access to the internet.

Research Process

This study was conducted within the scope of the "Preschool Science and Mathematics Teaching" course offered in the fourth term of the child development program. In the first four weeks, we provided participants with training on how to use Scratch. In training, we held discussions on the purpose and significance of Scratch and showed participants how to use the blocks (motion, sound, looks, etc.), add a new sprite, stage, and sound to a project, move the sprite, change the motion of the sprite based on a condition, and change the looks of the sprite based on a condition. In the following ten weeks, each participant created five science and five math Scratch projects about the preschool topics and learning outcomes of their choosing. Participants drew rough sketches describing the topics, learning outcomes/indicators, age groups, and features (sprites, sounds, etc.) they intended to address in their Scratch projects. They presented their drafts in front of the class, discussed their projects, and received feedback from their peers and teachers. They revised their draft projects based on feedback and moved onto their actual Scratch projects. One week later, they presented their Scratch projects in front of the class and received feedback from their peers and teachers. Table 1 shows the process.

Table 1. Research process

WEEKS	Learning Outcomes
1	Talking to participants about the purpose and importance of Scratch Introducing the Scratch interface and its features - Let's get to know the Scratch interface *Menu, stage, file, edit, stage ve backdrops, sprites, costumes, code, sounds*
2	Introducing the Scratch interface and its features Block Palette *Learning the functions of motion blocks* *Learning the functions of looks blocks* *Learning the functions of sound blocks* *Learning the functions of pen blocks* *Learning the functions of blocks in control block*
3	- Introducing the Scratch interface and its features. Block Palette *-Learning the functions of sensing blocks* *-Learning the functions of variable blocks* *-Learning the functions of operator blocks* -Showing participants the steps of a Scratch Project
4	- Let's explore the blocks *Participants created projects with ten blocks each in 15-20 minutes and then discussed them in class.*
5	-Creating Scratch I projects about preschool science topics -Creating design logs -Evaluating the Scratch projects and giving feedback
6	-Creating Scratch II projects about preschool math topics -Creating design logs. -Evaluating the Scratch projects and giving feedback
7	-Creating Scratch III projects about preschool science topics -Creating design logs. -Evaluating the Scratch projects and giving feedback
8	-Creating Scratch IV projects about preschool math topics -Creating design logs. -Evaluating the Scratch projects and giving feedback
9	-Creating Scratch V projects about preschool science topics -Creating design logs. -Evaluating the Scratch projects and giving feedback
10	-Creating Scratch VI projects about preschool math topics -Creating design logs. -Evaluating the Scratch projects and giving feedback
11	-Creating Scratch VII projects about preschool science topics -Creating design logs. -Evaluating the Scratch projects and giving feedback
12	-Creating Scratch VIII projects about preschool math topics -Creating design logs. -Evaluating the Scratch projects and giving feedback
13	-Creating Scratch IX projects about preschool science topics -Creating design logs. -Evaluating the Scratch projects and giving feedback
14	-Creating Scratch X projects about preschool math topics -Creating design logs. -Evaluating the Scratch projects and giving feedback

Tools for Data Collection

Another data collection tool used in studies is student studies. Student studies can be used as a data source (Johnson, 2014). Student studies are data showing the progress and development of students in the process (Hubbard & Power, 1993, as cited in Cavkaytar, 2009). Data was collected from participants' design logs and Scratch projects. Scratch is a project of the Lifelong Kindergarten (2003) group at the Massachusetts Institute of Technology (MIT) Media Lab. It provides a visual programming environment. It has many language options and appeals to a wide range of users (Çatlak, Tekdal & Baz, 2015). It allows the user to create projects by dragging and dropping without having to write down any codes. The code blocks can go on top of each other or be arranged side by side (Demirer & Sak, 2016). Creating design logs: *Please write down your thoughts and project processes as well as your sources of inspiration throughout your experience. Please write down your questions and their answers before sketching out your designs.* A rich digital programming tool, Scratch is supported by graphics, audio, and video. Thanks to the blocks in Scratch, they can be kept separate or combined to create the desired movement and interaction. Various concepts (geometric and measurement) are used with measurements such as coordinates, angles, and length that can be used to scratch online or offline. In this study, students created their projects using an offline program (Scratch 1.4).

Data Analysis

Thematic content analysis was carried out in the study. In the study, the main themes were determined and coding criteria were created. For the scratch projects created by the students in the study; 7 themes were created: subject, number of characters, appearance, movement, control, other blocks, and total number of blocks. The evaluation of scratch projects for the determined themes is given below (table 2). Validity, reliability, and coder reliability were assessed using the agreement percentage proposed by Miles and Huberman (1994). Coding was done by the researcher and an independent researcher who is an expert in the field. A consensus of 89% was calculated among the coders.

Table 2. An example of an analysis of a scrtach project in the study

Theme	Subject	Number of Characters	Movement	View	Control	Other Block	Total Number of Blocks
Code	Science subject (Buoyancy)	"Number of characters" used in the science subject (Buoyancy)	"going to x-y position" in the science subject (buoyancy)	"tell (..)" in the science subject (buoyancy)	When "clicking" in the science subject (buoyancy)	In the science subject (buoyancy) "recording… until you turn it off"	In the science subject (buoyancy) "total block usage count"

Results

This section addressed the results and presented them in tables.

Participants preferred to focus on different science and math topics. They had a wider range of selections regarding science topics than math topics (Table 3). They created Scratch projects about buoyancy (N=11), density (N=9), formation of rain (N=8), and mixtures (N=7). Some participants focused on germination (N=6), expansion (N=6), sense organs (N=6), and the importance of water for flowers (N=6). Table 3 shows the topics participants integrated into their Scratch projects.

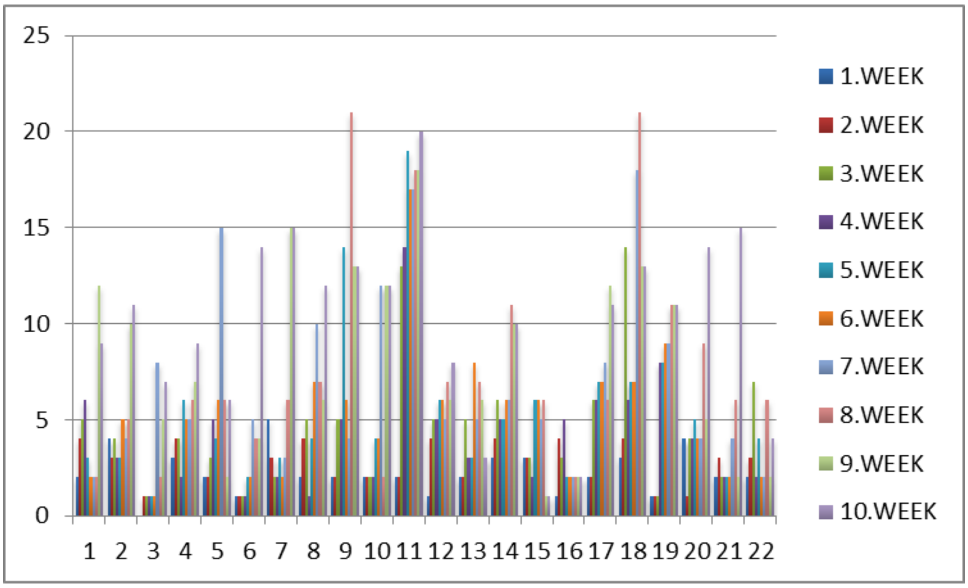
Table 3. Scratch topics

FIELD	TOPICS	N
SCIENCE	Seasons	1
	States of matter	2
	Buoyancy	11
	Refraction	1
	Gases	4
	Mixtures	7
	Germination	6
	What kind of materials does a magnet attract?	2
	Pressure	1
	Let's Get to Know the Animals	2
	Harmful Foods	1
	Formation of rain	8
	Expansion (air expands when heated - non-exploding balloon)	6
	Density	9
	Growing taller	1
	Importance of water for flowers	6
	Photosynthesis	1
	Taste buds on the tongue	2
	Combustion	3
	Metamorphosis of a butterfly	6
	Dissolution	2
	Sense organs	6
	Germs	5
	Growth of a fish	1

	Growth of a frog	1
	Volcanic eruptions	3
	Day-night formation	2
	Whirlwind	1
	Sound	1
	Growth of a bee	1
	Recycle	3
	Benefits of the Sun	1
	Harms of sugar	1
	Dental health	1
	Smell	1
MATH	Long-Short	2
	Matching	18
	Pattern	6
	Geometric shapes	15
	Full-Half	6
	Numbers	26
	Addition and subtraction	13
	Big-Small	13
	Heavy-Light	1
	Classification	1
	Counting	6
	Thin-Thick	1
	Under-Over	1
	Ranking	1

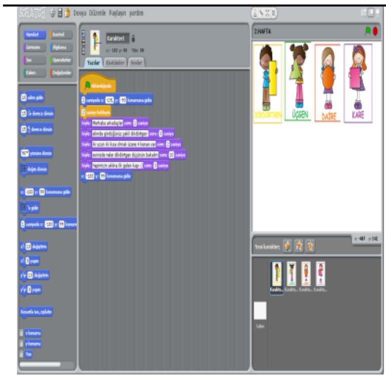
Participants created Scratch projects about 14 math topics. They mainly focused on numbers (N=26), matching (N=18), geometric shapes (N=15), addition-subtraction (N=13), and large-small (N=13). They also created Scratch projects about patterns (N=6), whole-half (N=6), and counting (N=6). Table 3 shows the science and math topics about which participants created Scratch projects.

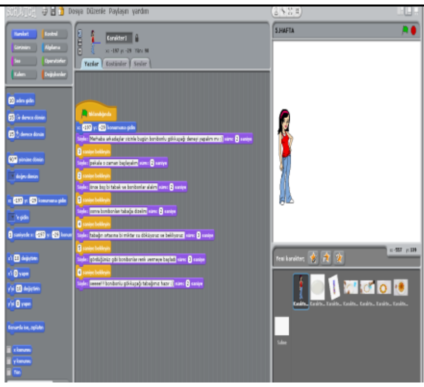
Graph 1 shows the distribution of characters participants used in their Scratch projects. It was revealed that students were better in mathematics (at 2nd, 4th, 6th, 8th, and 10th weeks) than in science (1st, 3rd, 5th, 7th, and 9th weeks). In the following weeks, it will be seen that the number of characters used is better in mathematics subjects. The number of characters varied across weeks and participants. Some participants used more and more characters as the weeks passed, while others did not show linear progress. For example, participant number 1 used one or two characters in the first week but used more than ten characters in the tenth week. On the other hand, participant number 11 used a sporadic number of characters throughout the weeks.

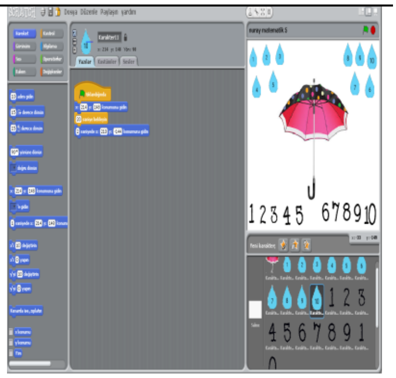


Graph 1. Number of Characters in Scratch Projects

Participants used the same number of characters in some weeks. Some participants used the same number of characters, especially towards the end of the competition. For example, participant number 16 used more characters in the first weeks but used the same number of characters towards the last weeks. On the other hand, Participant No 5 used the same number of characters in the first weeks, used more characters towards the end of the last weeks, and used fewer characters in the last weeks (Graph 1). You can see below the Scratch projects created by one of the participants: Number 18 in the second, fifth, and eighth weeks.

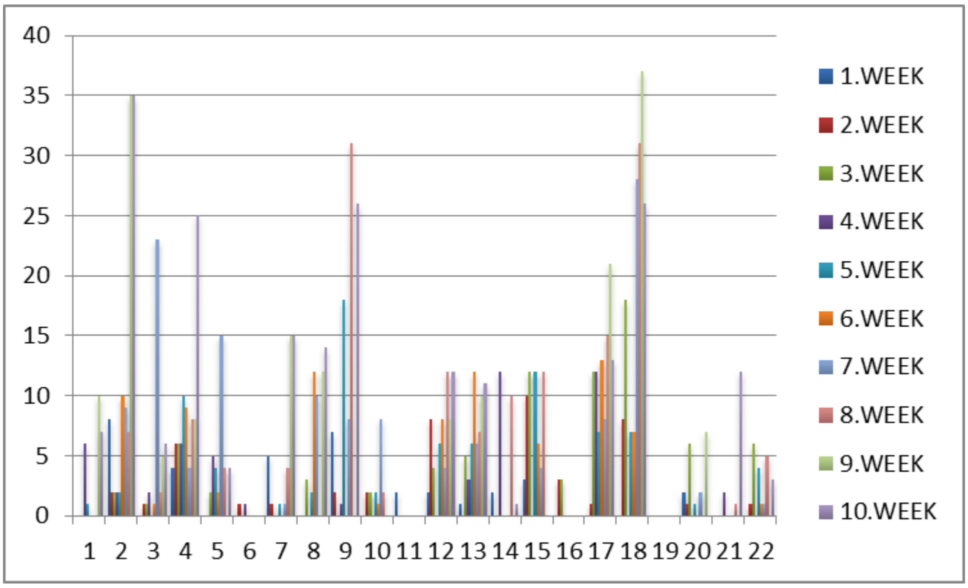






2.WEEK	5.WEEK	8.WEEK

The findings obtained from the study showed that there was an improvement in the frequency of use of the motion block in the field of mathematics (at the 2nd, 4th, 6th, 8th, and 10th weeks). There is a development in the frequency and content of motion block use (at 1st, 3rd, 5th, 7th, and 9th weeks) in science subjects. Participants did not use motion blocks very much. Some participants used motion blocks more often as the weeks passed, but some used very few motion blocks in some weeks. They mostly used the following motion blocks: "glide ( ) secs to x: ( ) y: ( )," "move ( ) steps," "turn right (15) degrees," "turn left (15) degrees," "go to x: ( ) y: ( )," "change y by ( )," "if on edge, bounce," "direction," "repeat ( ) times," turn (90) degrees," and "points towards ( )." For example, participants number 9 and 18 used motion blocks much more towards the end of the training.

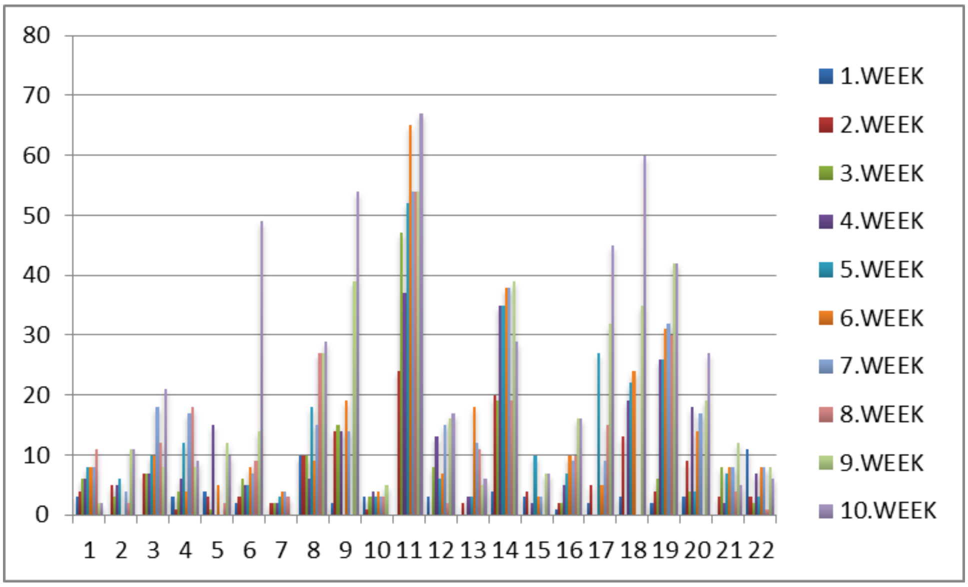


Graph 2. Frequency of use of motion blocks

Some participants never used motion blocks or used them very rarely in some weeks. For example, participant number 19 never used motion blocks, while participant number 21 used them only every two weeks (Graph 2). On the other hand, some participants used motion blocks very often. For example, participant number 18 used them very often in the first and last weeks. What is more, she has used them more in the last few weeks. You can see below the Scratch projects created by one of the participants: Number 2 in the second, seventh, and tenth weeks.

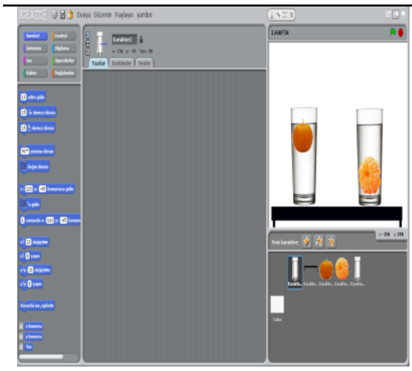
2.WEEK	7.WEEK	10.WEEK

Participants used the look blocks very often. They used them more and more in the last few weeks. When the findings obtained from the study are examined, it is seen that the frequency and content of the use of appearance blocks in the field of mathematics (at the 2nd, 4th, 6th, 8th, and 10th weeks) are enriched by weeks. In addition, in science subjects (at the 1st, 3rd, 5th, 7th, and 9th weeks), an increase is observed in the frequency of use of blocks by students. It is seen that the use of block content in mathematics and science subjects is similar. For example, Participant number 11 used the look blocks more often as the weeks passed and used them the most in the last week. They mostly used the following look blocks: *"say ( )," "add," "think ( ) for ( ) seconds," "show," "hide," "switch costume to ( )," "think (hmm)," "next characters," "change (color) effect by (25),"* and *"think ( )."*

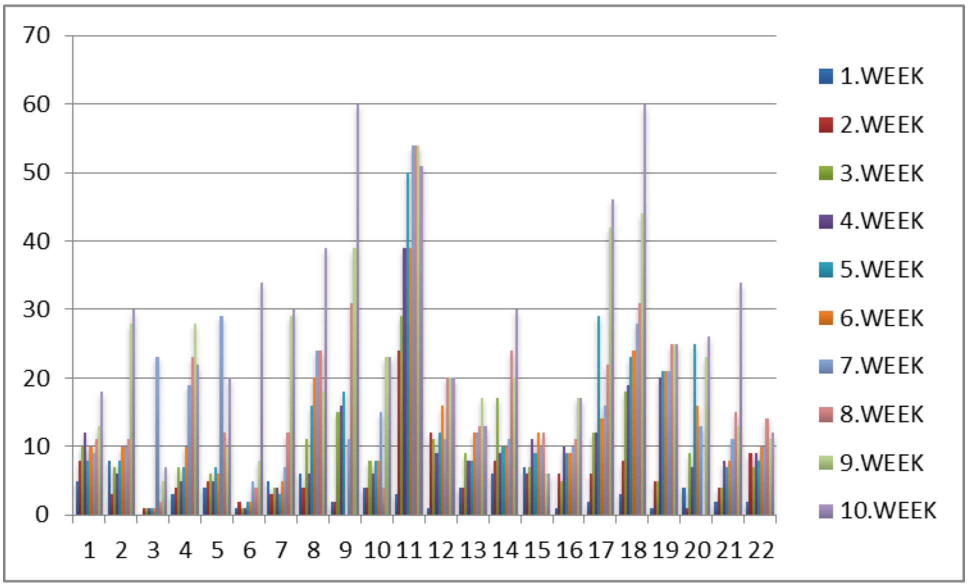


Graph 3. Frequency of use of the view blocks

Some participants made moderate levels of progress, while others made significant progress throughout the weeks. For example, participant numbers 1 made moderate progress, while participant numbers 7 made very little progress. Participants mostly used the look blocks of *"say ( ), "show,"* and *"hide."* You can see below the Scratch projects created by one of the participants: Number 11 in the first, fourth, and seventh weeks.

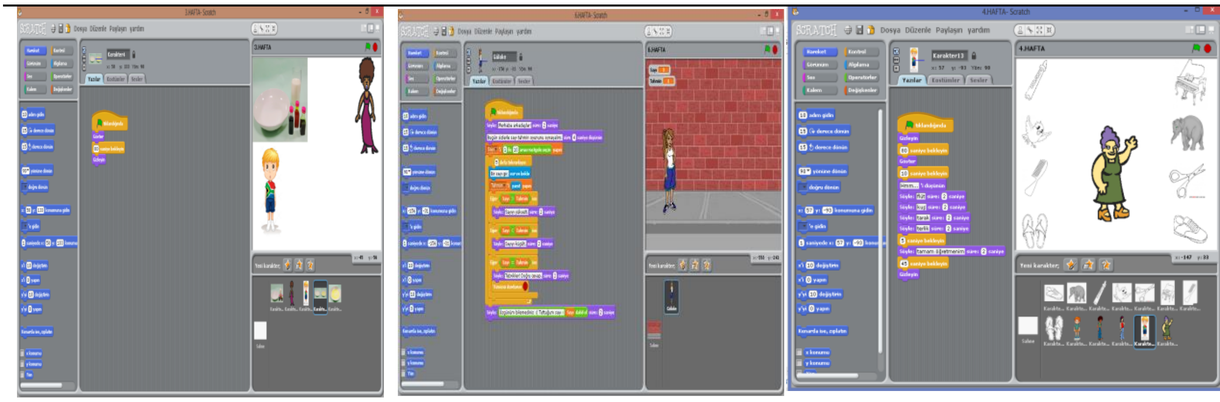
		
1.WEEK	4.WEEK	7.WEEK

Participants used control blocks often and used them more and more as the weeks passed. Not only did they use more control blocks, but they also diversified the content in the last few weeks. When the findings obtained from the study were examined, it was determined that the frequency and content of the control block usage became richer compared to the following weeks in the field of mathematics (at the 2nd, 4th, 6th, 8th, and 10th weeks). However, in science subjects (1st, 3rd, 5th, 7th, and 9th weeks), the frequency and content of block usage remained weaker than mathematics. They mostly used the following control blocks: *"when clicked," "wait ( ) seconds," "forever," "pause," "repeat ( )," "when sprite 6 clicked," "if ( ) then," "else," "direction," "when (key pressed)," "broadcast," "when I receive ()," "stop."*



Graph 4. Frequency of use of the control blocks

As for control blocks, some participants made significant progress, but others made moderate progress (Graph 4). For example, participant number 18 made consistent progress, while participant number 20 had ups and downs. Participant number 6 made significant progress in the last week. You can see below the Scratch projects created by one of the participants: Number 8 in the first, fourth, and tenth weeks.

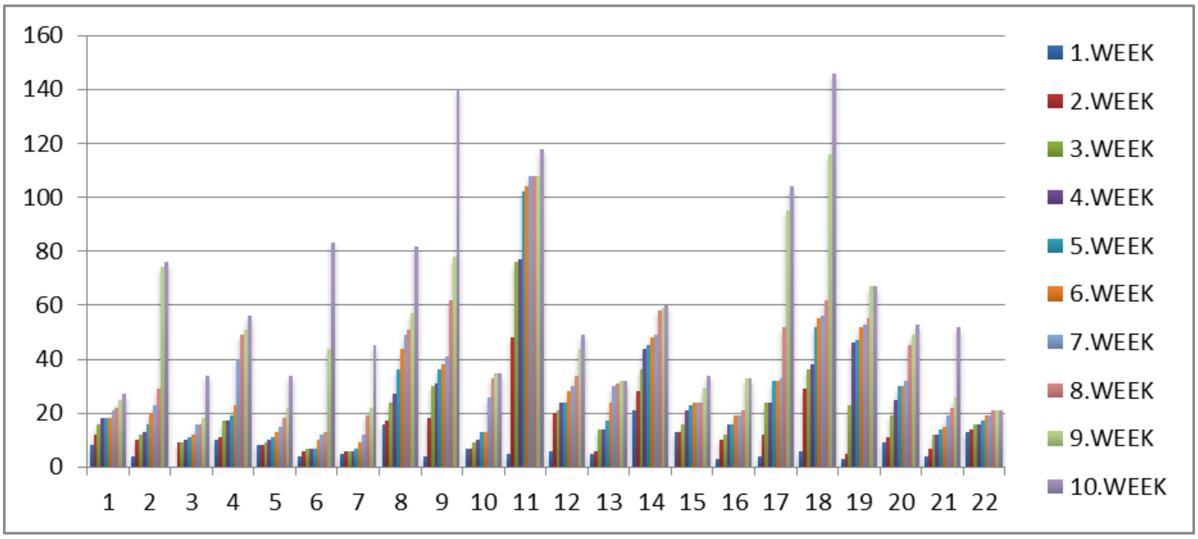


1.WEEK | 4.WEEK | 10.WEEK

Most participants made no progress in integrating other blocks (sound, variables, operators, etc.) into their projects. However, some participants used those blocks in their projects. It has been observed that students do not use other blocks very much in the fields of mathematics and science. However, it was determined that the frequency of use was better in the field of mathematics (at the 2nd, 4th, 6th, 8th, and 10th weeks). Eight participants (3, 8, 11, 14, 13, 16, 21, and 22) used sound blocks: *"recording, play sound... until done," "set volume to (100) %," "recording, play sound...,"* and *"play sound (soothing rain) until done."* Seven participants (1, 6, 8, 14, 15, 17, and 19) used sensing blocks: *"ask () and wait"* and *"answer."* Five participants (1, 3, 14, 15, and 19) used the operators blocks: *"pick random 1 to 10," "join," "()+()," "()=()),"* and *"()>( )."* Five participants (6, 8, 14, 15, and 19) used variable blocks: *"set the number of correct answers to ()," "set the number of incorrect answers to ()," "change the number of correct answers until ()," change the number of incorrect answers until ()," "Item 1," "conclusion," "Item 2," "number," "prediction,"* and *"number ()."* Only one participant used pen blocks: *"erase all," "set pen color to," set pen size to ()," and "pen down."* Below are participants' Scratch projects by weeks. You can see below the Scratch projects created by one of the participants: Number 14 in the second, fifth, and tenth weeks.

2.WEEK | 6.WEEK | 10.WEEK

The total number of blocks was the sum of eight different types of blocks. Participants used more and more blocks throughout the week. Some participants made progress in some weeks but not in others. However, most participants became better at using the blocks in their projects as the weeks passed (Graph 5). When the findings obtained from the study were examined, it was seen that the total block usage frequency of the students in the field of mathematics (2nd, 4th, 6th, 8th and 10th weeks) improved with the following weeks. On the other hand, in science subjects (1st, 3rd, 5th, 7th, and 9th weeks), it is seen that the progress is less compared to mathematics.



Graph 5. Frequency of use of the total blocks

Most participants used a few blocks in the first weeks but used more blocks in the following weeks. They peaked, especially in the last weeks. For example, participant number 2 used very few blocks in the first weeks but made significant progress as the weeks passed. Participants 6, 7, 13, 19, and 21 used the fewest blocks in their projects in the first weeks. On the other hand, participants number 9 and 18 used the highest number of blocks in the last week. Some participants made steady progress as the weeks passed. You can see below the Scratch projects created by one of the participants: Number 9 in the third, fifth, and tenth weeks.

3.WEEK 5.WEEK 10.WEEK

Conclusion, Discussion, and Suggestions

According to the results obtained from the study, the students increased the frequency of using blocks in scratch projects over time. It was observed that the content was enriched with the increase in the frequency of block use by the students. It is seen that the students are generally better in mathematics (at the 2nd, 4th, 6th, 8th, and 10th weeks) compared to the science subjects (1st, 3rd, 5th, 7th, and 9th weeks) with the following weeks. Begosso and Da Silva (2013) conducted a study on the use of Scratch to focus on improving students' (ranging from 11 – 13 years old) problem-solving skills and logical mathematical thinking. The students prepared an activity for three months by using the state, loop, and comparison operators in Scratch to improve their programming and problem-solving skills. The results obtained from the study revealed that the students showed improvement in these areas. This study analyzed students' Scratch projects. The results showed that participants used more and more blocks in their projects and integrated them into more diverse content. They focused on the subjects of buoyancy (science) and numbers (math) the most. Some participants used more sprites, while others did not make any significant progress. Participants had a consistent attitude towards motion blocks. Some participants used more motion blocks in the last weeks compared to the first weeks. However, some others did not make any progress. Some participants did not even use motion blocks in the last few weeks. Participants mostly used control and look blocks in the first weeks but made significant progress throughout the weeks. Participants had difficulty using control blocks and synchronizing their projects in the first few weeks. In the following weeks, they were able to use control blocks and had no problems with synchronization. Meerbaum-Salant et al. (2013)

argue that Scratch allows students to develop affective skills but fails to help them internalize some concepts (variables, synchronicity, and repetition). Only a few participants used the other blocks (sound, sensing, variables, and pen). The most important result was that participants used more blocks as the weeks passed. Although they used more or less the same blocks throughout the weeks, we can still talk about progress.

The reasons for these results obtained from the study can be explained as follows. First, this course was at the end of the semester. Second, participants probably had high stress and anxiety because they had to pass many courses to graduate. Third, the students had to take into account many factors because this was the first time they had had such an experience. In other words, students are supposed to have knowledge of many different fields. For example, they should know about technology for coding and have content knowledge for science and math concepts. They should have the pedagogical knowledge to teach these concepts to their students and to prevent misconceptions. They also need to have design skills to think about all these areas of knowledge together. All of these may be the reasons why participants had difficulty using Scratch.

Limitations of the Study and Recommendations for Future Researchers

The results of the study showed that in the analysis of the projects prepared by the students in science and mathematics, there were improvements in the frequency of use of blocks by weeks and enrichment in their content. As the first limitation of our study, we can state the absence of experimental and control groups. While the results obtained show that there is progress in our study, having an experimental and a control group will help us examine the performance differences in the process. Another limitation is the sample size. Researchers who will do research in the future claim that block-based coding is easy, but we think it should take more than a semester to integrate Scratch into lessons. In addition, it can not be limited to the fields of science and mathematics but can also integrate it into different disciplines. In addition, longitudinal execution of studies in this field and planning a longer-term study can reveal whether students benefit from this application.

Ethical Approval

The data used in this study was confirmed by the researcher that it belongs to the years before 2020.

Reference

Adler, R. F., & Kim, H. (2018). Enhancing future K-8 teachers' computational thinking skills through modeling and simulations. *Education and Information Technologies*, *23*(4), 1501-1514.

Aivaloglou, E., & Hermans, F. (2016). How kids code and how we know: An exploratory study on the scratch repository. *Paper presented at the Proceedings of the 2016 ACM Conference on International Computing Education Research*, Melbourne, VIC, Australia.

Altanis, I., & Retalis, S. (2019). A multifaceted students' performance assessment framework for motion-based game-making projects with scratch. *Educational Media International*, *56*(3), 201-217.

Atmatzidou, S., & Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, *75*, 661-670.

Balanskat, A., & Engelhardt, K. (2015). Computing our future. Computer programming and coding. Priorities, school curricula and initiatives across Europe. *European Schoolnet, Brussels*.

Bau, D., Gray, J., Kelleher, C., Sheldon, J., & Turbak, F. (2017). Learnable programming: blocks and beyond. *Communications of the ACM, 60*(6), 72-80. doi:10.1145/3015455

Begosso, L. C. & Da Silva, P. R. (2013*)*. Teaching computer programming: A practical review. *IEEE Frontiers in Education Conference (FIE),*508-510. https://www.computer.org/csdl/proceedings-article/fie/2013/06684875/12OmNyO8tUl (Accessed 10 March 2017).

Berg, B. L. (2001). *Qualitative research methods for the social sciences* (4th edition) [Electronic Version]. Allyn & Bacon Yayınları.

Bogdan, R. C. & Biklen, S. K. (2007). *Qualitative research for education: An introduction to theories and methods* (5th edition). Pearson Yayınları.

Brennan, K., & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking*. Paper presented at the American Education Researcher Association, Vancouver, Canada.

Buss, A., & Gamboa, R. (2017). Teacher transformations in developing computational thinking: Gaming and robotics use in after-school settings. In P. J. Rich & C. B. Hodges (Eds.), *Emerging research, practice, and policy on computational thinking* (pp. 189-203). Springer.
Calder, N. (2010). Using Scratch: An integrated problem-solving approach to mathematical thinking. *Australian Primary Mathematics Classroom, 15*(4), 9-14.
Cavkaytar, S. (2009). *Dengeli okuma yazma yaklaşımının Türkçe öğretiminde uygulanması: İlköğretim 5. sınıfta bir eylem araştırması* [Yayınlanmamış doktora tezi]. Anadolu Üniversitesi, Eskişehir.
Chao, P.-Y. (2016). Exploring students' computational practice, design and performance of problem-solving through a visual programming environment. *Computers & Educatifon,* 95, 202-215.
Çatlak, Ş., Tekdal, M., & Baz, F. Ç. (2015). The status of teaching programming with scratch: a document review work. *Journal of Instructional Technologies & Teacher Education*, *4*(3), 13-25.
Demirer, V., & Sak, N. (2016). Programming education and new approaches around the world and in Turkey/Dünyada ve Türkiye'de programlama eğitimi ve yeni yaklaşımlar. *Eğitimde Kuram ve Uygulama, 12*(3), 521-546.
European Commission (2014a). *Coding- the 21st century skill.* European Commission. https://ec.europa.eu/digital-single-market/coding-21st-century-skill
Fesakis, G., Gouli, E. & Mavroudi, E., (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education,* 63, 87-97.
Hendricks, C. (2006). *Improving schools through action research: A comprehensive guide for educators*. Pearson.
Johnson, A. P. (2014). *Eylem araştırması el kitabı* (Çev: Y. Uzuner ve M. Özten Anay). Anı.
Kalelioğlu, F. & Gülbahar, Y. (2014). The Effects of Teaching Programming via Scratch on Problem Solving Skills: A Discussion from Learners' Perspective. *Informatics in Education, 13(1),* 33-50.
Koh, K. H., Basawapatna, A., Nickerson, H., & Repenning, A. (2014, July). *Real time assessment of computational thinking*. In 2014 IEEE Symposium on Visual Languages and Human-Centric Computing (*VL/HCC)* (pp. 49-52). IEEE.
Korkmaz, Ö. (2018). The effect of scratch-and lego mindstorms Ev3-Based programming activities on academic achievement, problem-solving skills and logical-mathematical thinking skills of students. *MOJES: Malaysian Online Journal of Educational Sciences*, *4*(3), 73-88.
Kwon, K., Lee, K., & Chung, J. (2018). Computational Concepts Reflected on Scratch Programs. *International Journal of Computer Science Education in Schools*, *2*(3), n3.
Lewis, C. M., & Shah, N. (2012, February). *Building upon and enriching grade four mathematics standards with programming curriculum.* In Proceedings of the 43rd ACM technical symposium on Computer Science Education (pp. 57-62).
Lifelong Kindergarten Group (2003). *Scratch, MIT Media Lab*. Available from: http://scratch.mit.edu
Malan, D. J., & Leitner, H. H. (2007). Scratch for budding computer scientists. *ACM Sigcse Bulletin*, *39*(1), 223-227.
Maloney, J.H., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. (2008). Programming by choice: Urban youth learning programming with Scratch. *ACM SIGCSE Bulletin,* 40(1), 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), 1-15.
Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2013). Learning computer science concepts with scratch. *Computer Science Education*, *23*(3), 239-264.
Miles, M. B., & Huberman, A. M. (1994). *Qualitative data analysis: An expanded sourcebook*. sage.
Mills, G. E. (2003). *Action research: A guide for the teacher researcher* (2.Baskı). Merrill Prentice Hall.
Monroy-Hernández, A., & Resnick, M. (2008). Empowering kids to create and share programmable media. *Interactions*, *15*(2), 50-53.
Moreno-León, J., & Robles, G. (2015, March). Computer programming as an educational tool in the English classroom a preliminary study. In *2015 IEEE global engineering education conference (EDUCON)* (pp. 961-966). IEEE.
Moreno-León, J., Robles, G., & Román-González, M. (2015). Dr. Scratch: Automatic analysis of scratch projects to assess and foster computational thinking. RED. *Revista de Educación a Distancia, (46),* 1-23.
Moreno-León, J., Robles, G., & Román-González, M. (2016). Code to Learn: Where Does It Belong in the K-12 Curriculum? *Journal of Information Technology Education*, 15, 283-303.
Moreno-León, J., Román-González, M., Harteveld, C., & Robles, G. (2017, May). *On the automatic assessment of computational thinking skills: A comparison with human experts.* In Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems (pp. 2788-2795).

O'Brien, R. (2001). An overview of the methodological approach of action research [Elektronik Versiyon]. R. Richardson (Editor). *Theory and Practice of Action Research*. http://www.web.ca/~robrien/papers/arfinal.html

Oluk, A., & Korkmaz, Ö. (2016). Comparing students' scratch skills with their computational thinking skills in terms of different variables. *Online Submission*, *8*(11), 1-7.

Otrel-Cass, K., Forret, M., & Taylor, M. (2009). Opportunities and challenges in technology-rich classrooms: Using the Scratch software. *Set: Research Information for Teachers (Wellington)*, (1), 49-55.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books.

Peppler, K. A., & Kafai, Y. B. (2006). C*reative codings: Investigating cultural, personal, and epistemological connections in media arts programming.* https://repository.isls.org/bitstream/1/3652/1/972-973.pdf

Resnick, M. (2007, June). *All I really need to know (about creative thinking) I learned (by studying how children learn) in kindergarten*. In Proceedings of the 6th ACM SIGCHI Conference on Creativity & Cognition (pp. 1-6).

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.

Resnick, M., & Silverman, B. (2005, June). *Some reflections on designing construction kits for kids.* In Proceedings of the 2005 conference on Interaction design and children (pp. 117-122).

Sáez-López, J.-M., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools. *Computers & Education*, 97, 129-141.

Sant., J.A., (2009) *Mailing it in: email-centric automated assessment*. ITiCSE '09: Proceedings of the 14th annual ACM SIGCSE Conference on Innovation and technology in computer science education, 308–312, New York, NY, USA. ACM.

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

Tan, W. L., Samsudin, M. A., Ismail, M. E., & Ahmad, N. J. (2020). Gender differences ın students' achievements ın learning concepts of electricity via steam ıntegrated approach utilizing scratch. *Problems of Education in the 21st Century*, *78*(3), 423.

Taylor, M., Harlow, A., & Forret, M. (2010). *Using a computer programming environment and an interactive whiteboard to investigate some mathematical thinking*. Procedia-Social and Behavioral Sciences, 8, 561-570.

Tijani, F., Callaghan, R., & de Villers, R. (2020). An Investigation into Pre-service Teachers' Experiences While Transitioning from Scratch Programming to Procedural Programming. *African Journal of Research in Mathematics, Science and Technology Education*, *24*(2), 266-278.

Tseng, S. S., & Weng, J. F. (2009). Wiki-based design of Scientific Inquiry assessment by game-based Scratch programming. *Advanced Learning Technologies, 2009. ICALT 2009. Ninth IEEE International Conference*, 704-706.

Van Zyl, S., Mentz, E., & Havenga, M. (2016). Lessons learned from teaching Scratch as an introduction to object-oriented programming in Delphi. *African Journal of Research in Mathematics, Science and Technology Education*, *20*(2), 131-141.

Vihavainen, A., Vikberg, T., Luukkainen, M., & Pärtel, M., (2013). *Scaffolding students' learning using test my code*. In Proceedings of the 18th ACM conference on Innovation and technology in computer science education, 117-122.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., et al. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology, 25*(1), 127–147.

Wilson, A., Hainey, T., & Connolly, T. (2012). *Evaluation of computer games developed by primary school children to gauge understanding of programming concepts*. Proceedings of the 6th European Conference on Games-Based Learning (ECGBL), 4- 5.

Yildiz, S. N., Cobanoglu, A. A., & Kisla, T. (2020). Perceived Acceptance and Use of Scratch Software for Teaching Programming: A Scale Development Study. *International Journal of Computer Science Education in Schools*, *4*(1), 53-71.

Zhang, L. & Nouri, J. (2019). A systematic review of learning computational thinking through Scratch in K-9. *Computers & Education, 141* (2019), 1-25.

Zhang, H., Yang, Y., Luan, H., Yang, S., & Chua, T. S. (2014, November). *Start from scratch: Towards automatically identifying, modeling, and naming visual attributes*. In Proceedings of the 22nd ACM international conference on Multimedia (pp. 187-196).