(ISSN: 2602-4047)

Uluay, G. (2022). Programming Experiences of Pre-service Elementary School Teachers with Scratch, *International Journal of Eurasian Education and Culture , 7(17), 965-993.*

DOI: http://dx.doi.org/10.35826/ijoecc.453

Article Type (Makale Türü): Research Article

PROGRAMMING EXPERIENCES OF PRE-SERVICE ELEMENTARY SCHOOL TEACHERS WITH SCRATCH

Gülşah ULUAY
Assoc. Prof. Dr., Ordu University, Ordu, Turkey, gulsahuluay@gmail.com
ORCID: 0000-0002-6365-5122

Received: 06.08.2021 *Accepted: 10.05.2022* *Published: 15.06.2022*

ABSTRACT

Within the framework of the rapid change and development of technology, the emphasis on the importance of programming is increasing. It is stated that programming education supports individual development and learning processes in various fields such as 21st century skills. With this importance given to programming and its positive effect on learning processes, it is seen that programming courses are added to the curricula in many countries. The purpose of this study is to determine the views of pre-service elementary teachers about programming after the implementation process has been experienced with Scratch. This study was carried out with 54 senior students studying in the department of primary education at a public university in Turkey. The implementation process conducted on distance education system carried out 10 weeks. This study, which consists of qualitative data, was designed based on case study. Research data was collected through a structured interview form. The data collection process was carried out through interviews held at the beginning of the implementation process and at the end of the process. Two different interview forms were used, consisting of four open-ended questions in the pre-interview process and seven open-ended questions in the final interview process. According to the results of the research, it was determined that at the end of the implementation process, the participants mostly started to show a positive tendency towards programming. In addition, most pre-service teachers reported their desire to improve themselves in this field and that they plan to use programming technologies in their future professional lives.

Keywords: Programming, Scratch, teacher education.

INTRODUCTION

In recent years, it has been observed that many large companies such as Google and Microsoft have emphasized the importance of programming and teaching programming at the primary education level (Hainey, Baxter & Ford, 2020). As a matter of fact, programming is now considered a fundamental skill (Alturayeif, Alturaief & Alhathloul, 2020; Lindberg, Laine & Haaranen, 2019; Suzuki, Kato & Yatani, 2020). When the relevant literature is examined, it is seen that programming supports skills such as computational thinking (Resnick et al., 2009; Ruthmann, Heines, Greher, Laidler & Saulters, 2010), algorithmic skills (Chang, 2016), logical thinking (Bers, 2018; Fesakis & Serafeim, 2009), abstract thinking (Bers, 2018), problem solving (Bers, 2018; Chang, 2016; Fessakis, Gouli & Mavroudi, 2013; Fesakis & Serafeim, 2009), creative design process (Bers, 2018), social thinking (Fessakis, Gouli & Mavroudi, 2013) and creative thinking (Fesakis & Serafeim, 2009). Besides, it is stated that motivation develops with programming (Sáez-López, Román-González & Vázquez-Cano, 2016).

Programming environments offer alternatives such as text-based programming and block-based programming. In text-based programming environments, users use the syntax of the programming language (Göncü, Çetin & Şendurur, 2020). This syntax forces users who are at the beginning level of programming (Grover & Basu, 2017). Block-based programming, on the other hand, offers visual clues about how and where commands can be used in blocks in the form of puzzle pieces (Weintrop, 2019). Block-based programming environments are based on the principle that these blocks are dragged into the scripting area and combined to create a certain command (Weintrop & Wilensky, 2017). In this way, unlike text-based programming environments, syntax problems are eliminated, and users are allowed to focus on algorithms (Mladenović, Boljat & Žanko, 2018). Scratch, Alice, Snap!, App Inventor and Blockly are examples of block-based programming environments (Grover & Basu, 2017). Scratch has also gained a very popular place among these programming environments (Grover & Basu, 2017; Resnick et al., 2009). When the relevant literature is examined, it is seen that there are studies that examine the effects of the Scratch programming environment on various variables and report positive results. For example, in a study conducted by Hainey, Baxter and Ford (2020) in 16 different classes with 384 students aged 8-11, an implementation process in which the games-based construction learning (GBCL) approach and the Scratch programming environment are integrated was designed. Researchers have found that programming concepts can be taught effectively using Scratch at all primary education grade levels. In another study conducted by Sáez-López, Román-González and Vázquez-Cano (2016), Scratch integration was provided in the classroom environment with 107 students studying at five different schools at the 5th and 6th grade levels. The implementation process continued for two academic years. At the end of the implementation process carried out within the framework of project-based learning, the researchers emphasized the important effects of the programming environment on usefulness and motivation in line with the findings they obtained.

With the rapid development in the field of technology (Ashrafzadeh & Sayadian, 2015) and spread of technology (Dolenc & Aberšek, 2015) in many areas, changes have occurred in various aspects of life such as working, learning and communication (Noeth & Volkov, 2004). Resnick et al. (2009) stated that most of the young people

are very easy to send messages, play online games and use the Web, but they explained that these situations are not an indication of familiarity with new technologies. As a matter of fact, researchers state that young people are constantly interacting with digital media, but very few of them can design their own games, animations, or simulations. In addition to this situation, it is stated that many students think that programming is difficult (Attard & Busuttil, 2020; Chang, 2016; Fesakis & Serafeim, 2009; Sheard, Simon, Hamilton & Lönnberg, 2009). In this context, it can be stated that it is important to start programming education at an early age to raise individuals who both use technology effectively and do not have negative prejudices towards such technologies. When the relevant literature is examined, it is seen that the importance of starting programming education especially from the elementary level is emphasized (Suzuki, Kato & Yatani, 2020). From this point of view, it is obvious that the pre-service elementary teachers who will give elementary level education in their future professional lives learn programming. At this point, in this study, a programming-oriented implementation process was designed for pre-service elementary school teachers. In this context, the purpose of this study is to determine the views of pre-service elementary teachers about programming after the implementation process, they have experienced with Scratch.

METHOD

In this study, case study design, which is one of the qualitative research types, was used. In the case study, examples of real situations experienced by individuals are presented (Cohen, Manion & Morrison, 2007). At this point, the concept of case is defined as the aim of the research (Stake, 1995). From this point of view, the case of this study can be expressed as determining the views of pre-service elementary school teachers about programming.

Participants

This study was carried out with 54 senior students studying in the department of elementary school teaching at a public university in Turkey. 44 of all participants were female and 10 of them were male. The average age of the participants was determined as 22.8, with range between the ages of 22 and 23. The criterion sampling method was used to determine the participants. According to this method, individuals, or groups to be included in the research are expected to meet the specified criteria (Omona, 2013). In this study, the criterion of the participants' successful completion of the information technologies course in the curriculum was considered. This course covers basic topics such as algorithms and flowcharts, computer systems, basics of operating systems, file management, word processing programs (Council of Higher Education, 2018). Besides, voluntary participation was taken as a basis and pre-service teachers who met the relevant criteria participated in the implementation process in line with their own wishes.

Data Collection Process

This study was carried out through two-hour sessions per week held within the scope of the 10-week implementation process. Due to the pandemic, the sessions where the distance education process was started

in Turkey were conducted online. Participants attended each session simultaneously. In addition, each session was recorded, and the participants were able to access the recordings of the session whenever they wished.

Scratch was used in the implementation process. Scratch is a free programming environment developed by MIT Media Lab (MIT Media Lab, 2021). This environment based on block-based programming supports its users with a wide variety of language options, including Turkish (Başarmak & Hamutoğlu, 2019). Also, the programming environment offered by Scratch allows users to create a wide variety of interactive and rich media projects such as simulations, games, science projects, music videos (Maloney, Resnick, Rusk, Silverman & Eastmond, 2010). Programming can be done with combinations of ready code blocks in the Scratch environment (Yoshihara & Watanabe, 2016).

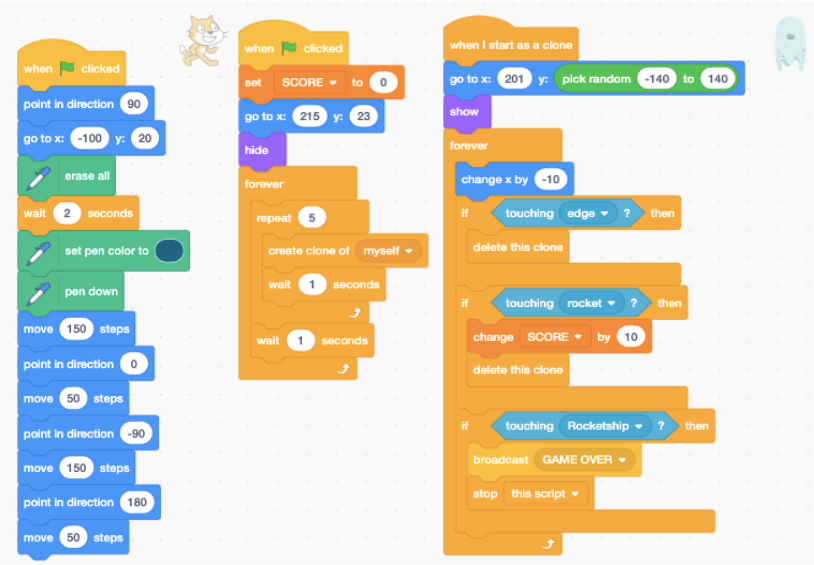

Figure 1. Examples of Algorithms with Scratch's Code Blocks

Explanations about the content followed during the implementation process are presented in the following table.

Table 1. Content of Data Collection Process

Week	Activity
1	Explanation of programming and algorithm concepts Introducing the home screen and tabs of Scratch Adding a sprite Adding and changing backdrops Creating and saving a project
2	Introducing code blocks, costumes, and sounds Coordinate system Creating a dialog
3	Variable operations Creating a counter
4	Designing animations with Scratch
5	Designing animations with Scratch
6	Designing games with Scratch and scoring
7	Designing games with Scratch and scoring
8	Participants make their own designs by taking the necessary support
9	Participants make their own designs by taking the necessary support
10	Participants present their designs General evaluation

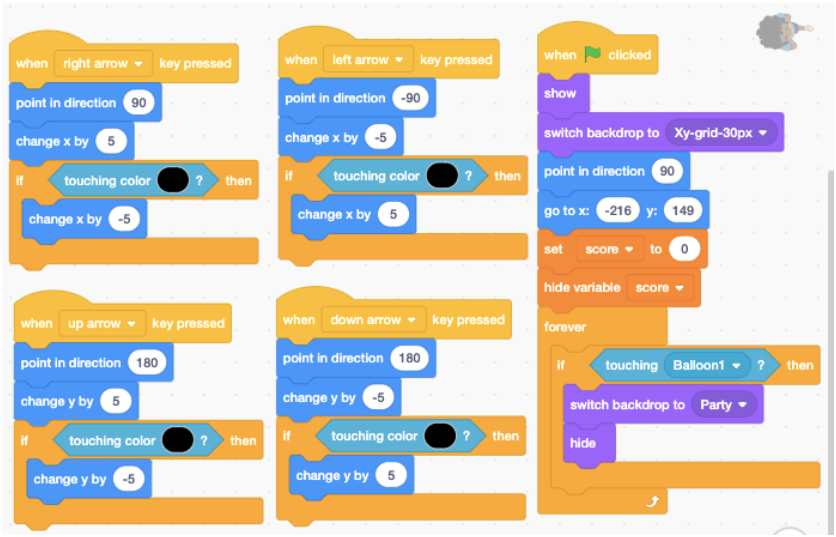

Figure 2. An example of Algorithms of a Participant

Republic of Turkey Ministry of National Education carries out the design and updating of the curricula applied from the pre-school period to the higher education period in Turkey. The participants examined the science objectives within the framework of the specified elementary school curriculum and designed their own projects within the framework of these objectives. Selected science subject matters can be specified as follows: (1) Sense organs, (2) Living things and environment, (3) Adequate and balanced nutrition, (4) Environment and environmental control, (5) Recycling. Science subjects termed as sense organs, living things and environment, and environment and environmental control are included in the 3rd grade science curriculum. The other two subjects entitled as adequate and balanced nutrition, and recycling are emphasized in the outcomes of the 4th grade curriculum.

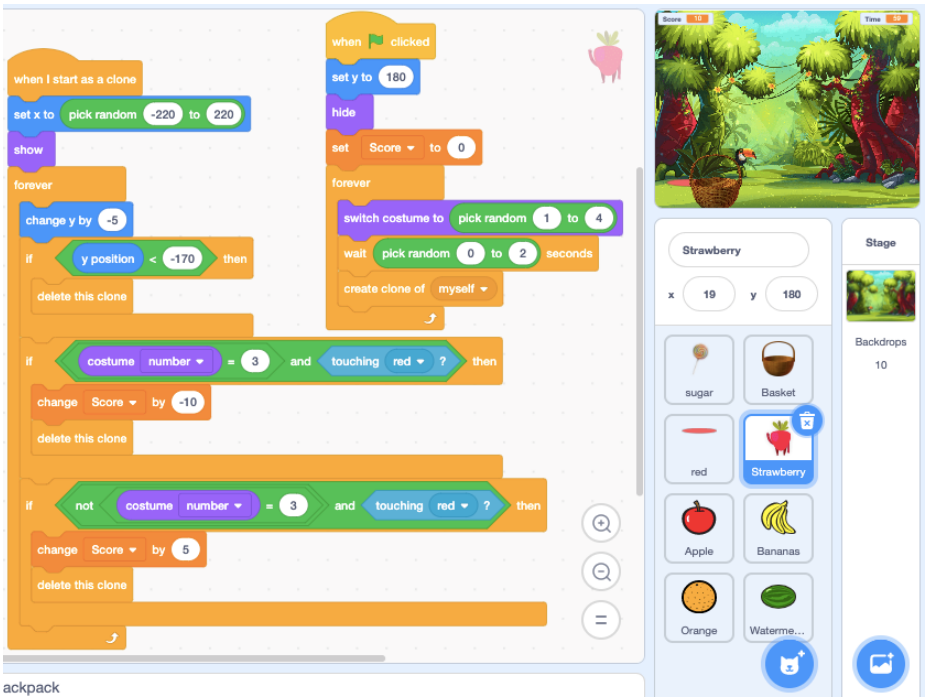

Figure 3. An Example of Participant Project

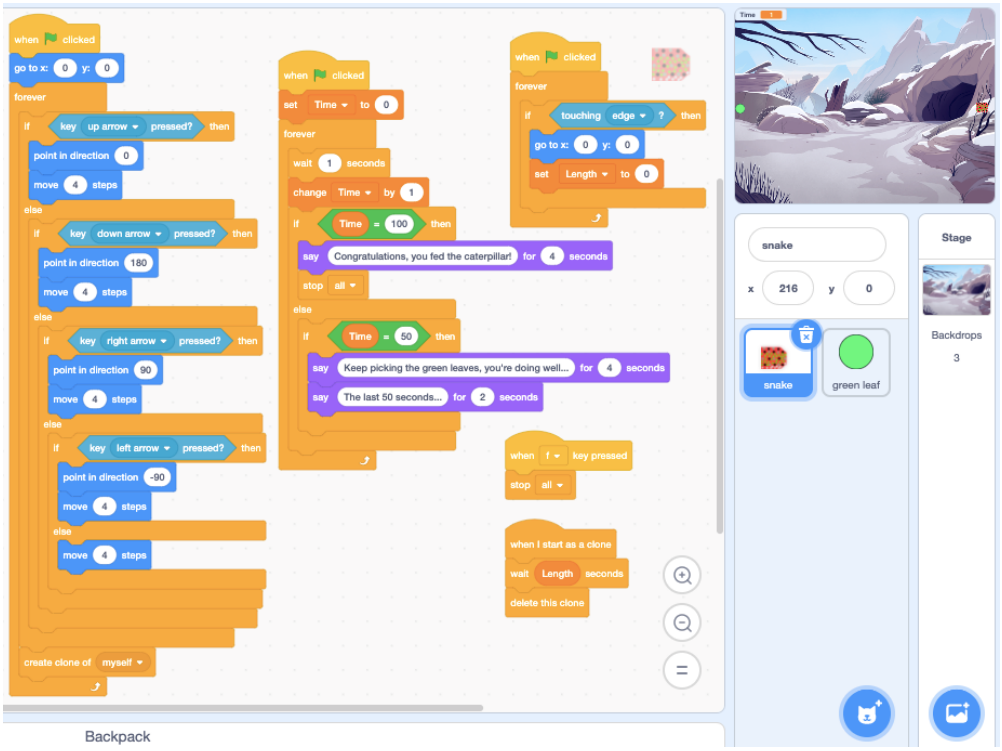

Figure 4. An Example of Participant Project

Data Collection Tools

Two semi-structured interview forms were used as data collection tools of this qualitative research. At the beginning of the implementation process, the pre-interview process was carried out to determine the existing views of the participants about programming. After the implementation content was completed, the post-interview process was conducted.

Table 2. Interview Questions

Process	Phase	Directions	Time
Beginning	Explanation	Provide information about the purpose of the interview Providing information about interview recordings and confidentiality of the study Requesting recording permission	5-6 min.
Pre-interview (Before implementation)	Questions	1. What is programming? What is your first thought when programming is mentioned? 2. Have you used any programming software/platform before? If you have used, please provide information about your experience. 3. Can you give an example of any programming software/platform you've only heard before? 4. Explain your thoughts about the role and effects of programming in present day.	15-20 min.

Post-interview (After implementation)	Questions	1. Please evaluate use of Scratch in terms of ease/difficulty. 2. Can you explain the points you have difficulty in learning how to use Scratch? 3. What is your favorite feature of Scratch? 4. What is your least favorite feature of Scratch? 5. Can you explain why you like programming or not? 6. Can you explain your views on the integration of programming practices into education? 7. Would you like to improve yourself in the field of programming? Why?	45-50 min.

Data Analysis

The qualitative data obtained from this study was analyzed based on flow model explained by Miles and Huberman (1984). Researchers stated that the qualitative analysis process carried out within the framework of this model consists of three activities as data reduction, data display and conclusion-drawing/verification carried out simultaneously (Miles & Huberman, 1984). Besides, to protect participant confidentiality in the presentation of the findings, pre-service teachers were named with the abbreviations determined as Pn (P1, P2, …, P54).

Strategies that can be used for dependability in qualitative research are identified as member checks, peer debriefing, triangulation, prolonged engagement in the field, persistent observations in the field, reflexive journals, and independent audits (Cohen, Manion & Morrison, 2007). In this study, peer debriefing was used in the processes of preparation of interview questions, data analysis and reporting. Researchers should regularly step away from their own perspectives and interact with other professionals who are willing to participate in the process for inquiry (Guba, 1981). This strategy which is termed as peer debriefing is the disclosure of the views of an unrelated expert to test the honesty and working hypotheses and determine the next steps in the research process (Cohen, Manion & Morrison, 2007).

FINDINGS

In this section, the findings regarding the data obtained in pre-interview and post-interview processes are presented respectively.

Pre-Interview Process

With the pre-interview process conducted before the implementation process, it was aimed to determine the participants' existing views about programming. In this context, the results of the content analysis regarding the data collected within the scope of four open-ended questions are presented in this section respectively.

Question 1. What is programming? What is your first thought when programming is mentioned?

When the explanations of the participants regarding the concept of programming were examined, five categories called individual perception, emotional impact, type, technique, and equipment were obtained.

Table 3. Participants' Views on Programming

Categories	Codes	n
Individual perception	The need of the age	28
	Important	25
	Popular	25
	Complex	27
	Difficult	26
	Requiring intelligence	5
Emotional impact	Scary	29
	Interesting	17
	Disturbing	15
	Exciting	2
Type	Computer programs	13
	Games	8
	Website	5
	Mobile apps	3
Technique	Issuing commands	15
	Various symbols	14
	Digital language	4
	Writing a computer program	3
Equipment	Computer	16
	Electronic devices	14
	Computer processor	3

The category termed as individual perception includes 6 codes which are called the need of the age, important, complex, difficult, popular, and requiring intelligence. Within the framework of this category, it is seen that the most emphasized view by the participants is the description of programming practices as the needs of the age. Similarly, 11 participants stated that today's world is living in the technology age and that programming is the future, and these implementations have a very important place. The emotional impact category consists of 4 codes termed as scary, interesting, disturbing, and exciting. It is seen that the most emphasized common view under this category is the feeling of fear created by programming. At this point, 9 participants explained their fear of programming within the framework of the thought of not having sufficient computer knowledge and skills. In addition, 10 participants stated that the thought of performing programming made them uneasy. While explaining the reasons for this situation, the participants expressed their views about the complex structure of programming processes and the necessity of advanced computer knowledge. 5 of the 17 participants who determined that they found programming interesting, underlined that they did not have any negative prejudices against these technologies. In the category termed as type, there are 4 codes named computer programs, games, websites, and mobile apps. The views within this category focus on the products that emerge because of programming practices. Within the scope of this category, the most emphasized views are that programming in general terms associate computer programs and more specifically computer games. The technique category created within the framework of the views focused on the operations performed in programming implementations includes 4 codes called issuing commands, various symbols, digital language and writing a computer program. Participants expressed their views about any action that is wanted to take place within the scope of the code called issuing commands and transforming it into commands. However, 14 participants reported that when they heard the programming word, they first thought of the screens containing letters, numbers or symbols that seem meaningless to them. In addition, 4 participants explained this concept as a digital

language by comparing programming to learning a new language. The responses of the participants who expressed their opinions within the framework of various technological environmental components, formed the category called equipment. This category includes 3 codes: computer, electronic devices and computer processor. The code named as computer represents the most emphasized view (n = 16) in this category. However, 14 participants reported that many electronic devices, other than computers, were products of programming implementations. Examples of participants' responses are presented below.

P1: "The name of programming has always reminded me of fear and confusion, but the work done through programming has always attracted my attention. The fact that it takes place in many systems tells how important programming is. I think that programming is the future in a sense. I can say that I am interested, except that it sounds complicated. To be honest, I am not good at computer. I can say that this situation scares me. I hope it's not as complicated as I thought, and I'm more interested."

P10: "Unfortunately, the first definition that comes to mind when programming is mentioned is the image formed by small meaningless text, numbers and symbols on the computer screen. Actually, I don't think I know much about technology. I'm not even a very active user of social media accounts. My computer skills are not that good either… It seems like a very complex, difficult subject to me. However, I will try to be as enthusiastic about learning as I can. I also think that programming should be used in education. We are in the age of technology, and I am starting from myself, if I had the chance to attend such classes in my primary school life, I would like computers, technological tools, etc.… My approach would be much more active and high quality. The world now revolves around computers. As much as I dislike it, I must participate in this transformation. Also, I personally do not want my students to be prejudiced."

P11: "When I think about programming, the first thing that comes to my mind is computer games. "What is programming?" If I think about the question in more detail: I can say that all devices are programmed for a purpose and the expected activities are provided by writing codes. I heard that programming is like learning a new language. That's why I don't have a negative opinion, on the contrary, I am excited…"

P16: "For me, programming is the name of the commands given to electronic devices and these devices that enable them to work in a certain order, from the computer we use to the telephone. When it comes to programming, I think of sentences full of letters and numbers that I can't understand now. I have prejudices against programming. I think only experts in this field can do it, and I think it's a really difficult thing."

Question 2. Have you used any programming software/platform before? If you have used, please provide information about your experience.

When the answers to this question were examined, it was seen that only 4 of the pre-service teachers participating in this study had previous experiences with programming practices. While 3 participants (P8, P9 and P15) stated that they used Scratch, 1 participant (P13) identified that he had experienced Algo Digital and

Code.org environments. However, 1 participant (P15) determined that he made implementations with Microsoft Visual Studio. 50 participants stated that they did not use any programming environment or program. Examples of experienced participants' responses are presented below:

P8: "I used Scratch in high school. There is an orange cat in its logo. Animations and games can be designed using the cat in this program… This easy-to-use program is suitable for beginners."

P9: "I used the Scratch program. In a way, this program is a graphical programming language… Everything consisted of codes. For each movement, a programming must be made. For example, for a person who wants to drink tea, which hand will be used first, the hand used will move in the coordinate plane, if there is any condition after holding the glass, it will be coded again for this, etc. We have to code the finest details of every action taken…"

P13: "I used Algo Digital. It was a program carried out by the voluntary education foundation I went to. Children were allowed to progress by writing the codes themselves that enabled them to reach the goal. It was a very important program in terms of development. We also worked with the Code.org environment as one-on-one with the children in the caravan of this voluntary education program. It was an efficient program that provided the development of children's hands and minds. It was happening gradually."

P15: "We had programming lessons in high school. In this process, we designed a simple game, piano, shopping site, etc. We studied with several programs such as Microsoft Visual Studio. I don't remember those programs very clearly… I also used the Scratch program last year at the university. I made a simple catch game. I used Tom and Jerry as characters. Jerry was the main character. Tom was coming straight from left to right, trying to catch Jerry. Jerry, on the other hand, was running away from Tom by jumping when I pressed the spacebar. The only problem was that Tom kept coming at the same timing frequency. I couldn't set it…"

When the answers are examined, it is seen that P15 provided the most detailed information about programming practice he had experienced. However, other participants were asked about their implementation content as a complementary question. At this point, the participants stated that they could not remember the use of the programs and the practices they performed very clearly. In addition to this explanation, P9 reported that he thought that he might have confused about his work.

Question 3. Can you give an example of any programming software/platform you've only heard before?

When the answers given to this question which was asked to determine whether the participants were aware of any programming language example were examined, it was seen that a wide variety of examples were presented. These examples are presented in the following table.

Table 4. Examples for Programming Environments of Participants

Example	Participant
Aldo Dijital	P35
Blockly Games	P7
Bitdegree*	P13
C	P27
C+**	P27
C++	P18
Code.org	P3, P7, P9, P28
Codecademy*	P2, P3, P7, P13
Code Hour**	P28
Compute It!	P4
Coursera*	P13, P36
Delphi	P18
edX*	P31, P36
Hopscotch	P9
HTML	P11, P14
Java	P8, P11, P14, P15, P16, P27, P33
Java Script	P18, P27
Khan Academy*	P17, P26, P35, P36
Lightbot	P4
MineCraft	P20
Phyton	P15
Scratch	P3, P7, P26
Treehouse*	P31
Tynker	P4
Udemy*	P31
Visual Basic	P18

* Examples not covered by the programming language
** Incorrect name

While stating the examples presented in the table, the participants identified that they only heard these names, did not use these programming languages or were not sure about these examples.

Question 4. Explain your thoughts about the role and effects of programming in present day.

When the answers given to this question were examined, the views of the pre-service teachers about programming formed two categories called negative perception and positive perception.

Table 5. Participants' Views on Role of Programming

Categories	Codes	n
Negative perception	Extinction of traditional professions	26
	Unemployment	21
	Excessive computer use	12
	Weapon production	8
	Harming eye health	4
Positive perception	Getting a profession / Finding a job	19
	Development of the country	10
	Making life easier	10
	Saving on time	6
	Professional success	3

The negative perception category focuses on the negative effects that pre-service teachers think that technological advances in the field of programming will create on human life. At this point, it has been seen that there are participants who stated that robots have started to replace humans due to the developments in the

field of programming and robotic implementations, and accordingly, there are robots in many sectors. For example, it is thought that robots are working in the production and packaging areas of many factories and this situation will become more common day by day, leading to problems such as unemployment and extinction of traditional professions. However, some participants were uneasy about the reflection of the developments in the field of programming on the defense industry and the production of weapons. In addition, it has been determined that the new generation has a strong interest in technology and that these implementations will cause disproportionate computer use or that the eye health will be negatively affected by the hours spent in front of the computer. The positive perception category, on the other hand, was shaped by the participants who thought that the developments in the field of programming would have much more positive effects on human life in the future. At this point, contrary to the unemployment perspective in the negative perception category, some participants stated that they believed that a wide variety of occupations would emerge in the fields of programming and robotics in the future and that employment in these fields would be quite high. In addition, it was stated that national development would be achieved by adopting the profile of a producing society with these practices. Similarly, it has been observed that there are participants who stated that the technological products that will arise with the developments in these areas will make daily life easier or that people will save time by not having to do many things. It has also been determined that there is a belief that individuals who improve themselves in the field of programming will be more successful in their profession even if they do not work in this field. Examples of participants' responses are presented below.

P14: "… I think it has bad effects as well as benefits. I think it will lead to the end of traditional professions. For example, if there is no such thing as a cashier soon, I will not find it odd, or if I think for my own profession, a simple system can be set up and most teachers can be left without work. Of course, in addition to these, new professions related to technology will emerge, but this will not save unemployment. A software developer can build a system that does the work of hundreds of employees, and robots will start doing the work that humans would do. When I think about it, it starts to sound ridiculous to me, robots will work instead of humans and even robots will start doing everything for us. This situation makes me guess that machines will live life instead of us, and we will live in a dystopia. Although this thought is distant for most people right now, it seems like we are going to surrender our lives, our lives, when we say let's integrate technology into our lives…"

P19: "… programming now provides important job opportunities to people today. No matter what profession we choose in the future, we will be expected to master and develop the technologies we use. Programming will have an important place in this field. When we look at it from another perspective, when we write the jobs of the future on Google, programming and software come first…"

Post-Interview Process

Purpose of the post-interview process was to investigate participants' views about programming after implementation process. In this context, the results of the content analysis regarding the data collected within the scope of seven open-ended questions are presented in this section respectively.

Question 1. Please evaluate use of Scratch in terms of ease/difficulty.

When the participants' evaluations of the difficulty of the Scratch programming language they experienced were examined, 4 codes were obtained, which were named as first difficult then easy, easy, difficult, and depending on the design type.

Table 6. Views of the Participants on the Difficulty of Scratch

Codes	n
First difficult then easy	22
Easy	18
Difficult	10
Depends on the design type	2

Two participants (P15, P24), who attributed the difficulty of the Scratch programming language to the type of design, reported that this situation changed depending on the project they were working on. The participants in question stated that the type of product developed (P24) and the coded action (P15) affected the level of difficulty.

P15: "… It was a lot of fun while programming my sprite. I got it done with movement and control. I had a hard time programming my basketball. Because when it touched the princess, I couldn't adjust the score part. I chose the princess as the sprite. For example, I coded it very easily. But I had a hard time scoring…"

P24: "I found making animations quite easy. I've never had any trouble with it. But I had some problems with the commands while making the game. Code lines were very difficult for me in the game. The game is based on more active animation, so I think it is necessary to create more complex codes…"

The participants who explained the difficulty regarding the use of Scratch in the framework of the codes first difficult then easy, easy, and difficult, focused on various reasons while expressing their opinions. At this point, the codes created within the framework of these reasons affecting the opinions of the participants are presented in the following table.

Table 7. Participants' Views on Role of Programming

Categories	Codes	n
First difficult then easy	Practicing	14
	Recognizing code blocks	13
	Generating code lines	10
	Complex interface	8
Easy	Clear interface	15
	Language option	11
	Ready code blocks	9
	Clear guidelines	4
	Easy setup	4
	Online use	3
Difficult	Complex code blocks	7
	Interface	5
	Wide coverage	4
	Mobile application	3
	Generating code lines	3

Within the scope of the first difficult then easy category, there are participant responses stating that they had difficulties in the implementation of programming practices with Scratch at the beginning, but that this programming environment became easy to use as the process progressed. When this category is examined, it is seen that the view most emphasized by the participants is to practice. At this point, the participants expressed their opinions that the programming language, which they initially perceived as difficult, was easy after lots of practice. However, some participants stated that recognizing code blocks, creating lines of code, and understanding the interface which seems complicated by gaining familiarity with the interface needed time and practice, and therefore Scratch was difficult at first. The participant responses in this category emphasized that they realized that they were wrong at the beginning with the progress of the process and the increasing variety of practices. The category named Easy was created in line with the opinions of the participants who stated that they had no difficulties while using Scratch. In this context, the participants explained that they could easily perform programming operations due to the features of Scratch having a simple, plain, and understandable interface, offering Turkish language support, and having ready-made code blocks. In addition, they stated that they can easily install Scratch on their computers and that it offers online use and that it is easy to access the programming environment. However, participants who gave their views within the framework of the explicit instructions code stated that they were guided by the regular rows of code blocks and their protruding (puzzle-like) or non-protruding images. A difficult category was created in line with the responses of the participants stating that they had difficulties while using Scratch throughout the implementation process. Participants who reported the appearance of complex code blocks in this category explained that the blocks caused confusion. In addition, while 3 participants explained that this programming environment has a complex interface, 2 participants stated that there is a wide variety of things that can be done with Scratch and that it creates a challenging environment due to its wide scope. In addition, 2 participants (P2, P48) stated that they wanted to program with a smart phone, but they had great difficulty in this process. Examples of participants' responses are presented below.

P5: "Scratch was not a difficult program for me to use. It even sounded simple and plain. The variety in terms of characters and the range may be wide, but despite this, the implementation part was very comfortable. The instructions were very clear and the programming we were going to do fit together like a jigsaw puzzle or Lego piece. This gave me a clue as to what step to take and where to place it…"

P12: "It seemed very difficult at first. I felt very lacking in subjects such as computers and programming. However, as I created my project, I improved myself. I mean, I never thought that these applications would improve my computer usage so much. Making games, programming started to become good and fun… I enjoyed it…"

P18: "It was difficult for me. Code blocks were confusing. The programming was also very confusing. No matter how hard I tried in the future, I had difficulties in every project I wanted to design."

P25: “The Scratch program was difficult for me. Because it is a very comprehensive program and I was in the new learning phase, I had a little trouble exploring it. For example, the program has too many tabs. He also makes the program complicated.”

P43: “After I got over my inexperience, I understood how to use it. I found it very convenient, simple and plain to set up and use.”

Question 2. Can you explain the points you have difficulty in learning how to use Scratch?

This question was used to determine the problems and difficulties faced by pre-service teachers who experienced the stages of getting to know Scratch and programming with this programming language within the scope of the implementation process carried out in this study. 11 participants (for example, P8, P16, P48) reported that they did not encounter any difficulties while learning the use of Scratch and that they learned this programming language easily. The difficulties experienced by 43 participants in the process formed 4 categories called programming, algorithm, content, and operation.

Table 8. Activities which Participants Have Difficulty in Scratch

Categories	Codes	n
Programming	Nested code blocks	14
	Operators	10
	Switching backdrop	5
	Coordinate system	3
	Switching costumes	2
	Establishing relationships between sprites	2
Algorithm	Generating code line	24
	Show / Hide	6
	Selecting code block	3
Content	Finding the code block	23
	Finding a sprite	15
	Changing sprite properties	4
Operation	Adding sound error	3
	Save error	2
	Not giving error feedback	2

The programming category focuses on the difficulties that pre-service teachers encounter while performing programming processes. Pre-service teachers reported that they had difficulties especially in the operations performed in nested code blocks. For example, they stated that they had problems with programming with the "If, If Not" code block. However, some participants stated that they had difficulties in operator operations. On the other hand, the opinions of the pre-service teachers who had difficulty in programming the changes in the switching backdrop were included in the code of the backdrop transitions, while the answers from the participants who had problems in adjusting the x and y values on the coordinate plane were included in the code of the coordinate plane. Programming of costume transitions and the programming of situations such as characters touching each other are the least reported views of the category. The algorithm category, on the other hand, includes the opinions of the participants who have difficulties in determining the stages that need to be performed to reach the goal and creating these stages in order. It is seen that the most emphasized difficulty in

this category is generating the code line. The pre-service teachers explained the problems they experienced while creating the code lines by referring to examples such as providing movement, adjusting the conversation times, releasing news, gaining/losing scores, order of the code block and not realizing the desired action with the generated code. However, 3 participants reported that they had difficulty in choosing the code block while creating the algorithm. On the other hand, 6 participants mentioned that they had problems in understanding the logic of show/hide operations and the difficulty of creating algorithms with these operations. The difficulties experienced in the processes of recognizing the tabs of the Scratch platform constitute the scope of the category called content. At this point, the participants stated that they had difficulties especially in finding code blocks and finding sprites. However, 4 participants stated that they had difficulty using the tabs for sprite features such as changing the sprite's direction and color. Within the scope of the operation category, 3 participants stated that Scratch gave an error in the sound adding process, while 2 participants stated that they encountered an error during the project recording phase. However, 2 participants reported that it was difficult for them that Scratch did not give a guiding error feedback when the generated code line did not work. Examples of participants' responses are presented below.

P2: "Actually, it was a great advantage to be able to look back at the parts I forgot or skipped because the lessons were recorded, and I didn't have any difficulties. But still, the point that confused me the most in the beginning was the code-in-code part."

P9: "The code blocks and language were understandable. That's why I didn't have any difficulties…"

P21: "I had a hard time combining the codes and adding lines of code to the characters. There were many problems while adding the codes to the characters and the codes I added could not function. Then I tried to add lines of code again…"

P28: "I had a hard time playing the animation I made from start to finish…"

P34: "The order of placing the codes made it difficult for me. At the slightest code error, problems arise in the operation of the game or animation…"

P50: "At first, I had difficulties with the ordering of lines of code. I wondered which one should come first. But as I learned different lines of code, I was able to sort them more easily…"

Question 3. What is your favorite feature of Scratch?

When the answers of the pre-service teachers regarding this question were examined, 5 categories were obtained, namely functionality, audio visual factors, adding, design and platform. It was observed that only one participant (P26) stated that he did not have any favorite feature about Scratch.

Table 9. Favorite Features of Scratch

Categories	Codes	n
Functionality	Ready code blocks	23
	Easy to understand	12
	Drag and drop	7
	Language support	5
	Enjoyable	5
	Grouped code blocks	3
	Changing sprites' properties	3
	Changing the backdrop	2
	Painting sprites	2
	Various code blocks	2
Audio visual factors	Sprite options	10
	Backdrop options	7
	Sound options	3
	Variety of sprites' costumes	2
Adding	Sprite	7
	Backdrop	6
	Sound/music	6
Design	Creating the imagined design	18
	Animation design	14
	Game design	12
	Story design	6
Platform	Sharing opportunity	5
	Providing instructional resources	2
	Discussion forums	2

The functionality category is shaped around the features offered by Scratch and the user-friendly home screen views that the participants think are convenient for them. At this point, it was seen that the most emphasized opinion ready code blocks created positive thoughts towards Scratch and programming in the participants. 12 participants reported that the Scratch environment is understandable thanks to the easy-to-use main screen and its simple interface. Similarly, 7 participants stated that they enjoyed drag-and-drop programming. However, there are participant opinions expressing that they find the possibilities of changing the size, color and direction of the sprites and drawing the sprites entertaining. Similarly, it was determined that there were pre-service teachers who stated that they liked the backdrop change. The audio-visual factors category includes the responses of the participants who stated that they liked the content offered in Scratch libraries. The codes created in this category include participant responses expressing satisfaction with a wide variety of options for sprite, backdrop, voice and more than one costume of a sprite. The adding category was created in the light of participant opinions that Scratch allowed its users to add various file types selected from the computer or the internet to the programming environment. In this framework, it was seen that the Scratch environment, which is open to adding sprites, backdrop, and sound/music, was liked by the participants. The design category focuses on views on projects that can be done with Scratch. The most emphasized view in this category is the code of making the imagined design, which focuses on the positive participant emotions arising from the environment of Scratch, which allows its users to concretely visualize the design of their dreams without limiting. However, there are participant opinions expressing that they like to be able to design animation, game, and story if desired. The platform category includes participant opinions about the opportunities Scratch offers to its users on its website. At this point, the participants, who stated that Scratch supports project sharing and that these projects

set an example for them and that it is a great advantage to offer educational videos to its users for free, stated that they like these features very much. However, 2 participants stated that they enjoyed the discussion forums on the Scratch website. Examples of participants' responses are presented below.

P6: "One of my favorite features of the Scratch program is that it has a simple interface, is translated into many languages, treats all people equally, and offers educational lessons to its users…"
P22: "My favorite feature of Scratch is that we can program our own game, create our own animations, and make beautiful, different projects with our creativity. We use blocks to write code in Scratch. Being able to create my projects quickly and easily was another favorite feature as the block system is based on drag and drop method. Because I liked that it was both simple and enjoyable."
P31: "… Turning imagination into reality."
P32: "I loved making animations. Because I like cartoons and animated movies a little more than regular movies, so I've always wondered. I loved it because I learned how it works and how we can easily do it ourselves."
P40: "My favorite feature of the Scratch program is that the codes are grouped separately and therefore provides an easy-to-use feature..."
P46: "The ability to design my sprite myself was my favorite feature..."
P52: "It was very enjoyable to be able to choose the background very simply and combine the codes like a puzzle. I also loved the category-category classification of the codes. It was also very good that it was translated into Turkish…"

Question 4. What is your least favorite feature of Scratch?
When the answers given to this question were examined, it was seen that 19 participants (for example, P8, P14, P23, P46, P51) stated that they did not encounter a feature that they did not like in the Scratch environment. However, responses from other participants formed two categories: Expandable parts and disliked parts.

Table 10. Least Favorite Features of Scratch

Categories	Codes	n
Expandable parts	Sprites	23
	Backdrops	20
	3D	20
	Number of sprites' costumes	16
	Animated sprites	12
	Sounds	9
	Code blocks	9
	Full screen control	6
	Editing added images	5
	Number of instructional videos	4
Disliked parts	Lack of 3D	18
	Complex command operations	15
	No hint offered	12
	Require time	10
	Lack of mobile apps	8
	Adding sound	7
	Transition between backdrops	4
	Online save	3
	Export	3

The participants who made up the expandable parts category focused on the features they thought needed to be improved rather than the features that made them dislike Scratch. While explaining this situation, the participants mentioned that these features do not cause them to dislike Scratch, but that a richer environment will be obtained with these improvements. At this point, it was determined that suggestions were made to increase code blocks, sprites, sprites' costumes, backdrops, and sounds. It was also reported that the environment would be richer with a larger stage and the presentation of moving sprites. However, opinions were expressed in the direction of increasing the number of educational videos by diversifying them. It has been observed that there are opinions about increasing the toolbars used for editing the visuals added to the Scratch environment and improving this editing environment. The most emphasized answer within the category of disliked parts is the view that the environment should not be 3D. However, the fact that some of the commanding processes are quite complex and the practice does not provide clues during the algorithm creation phase are among the features that the participants did not like. On the other hand, 8 participants stated that they do not like programming on Scratch with a smartphone and the lack of a mobile application version. However, the participants, who stated that they had to divide the desired sound file during the sound addition stage and that they encountered situations such as rewriting the codes during the transitions between the backdrops and not performing the costume transitions, stated that they did not like these features. 3 participants stated that an error occurred while saving their online projects and they could not save their projects. Similarly, 3 participants reported that they encountered problems with the export of the recorded project. Besides, 10 participants explained that due to the complex structure of the Scratch environment, more time should be spent to understand it and they did not like this situation. Examples of participants' responses are presented below.

P1: "The worst thing about Scratch was that when there was a problem at some point in the animation or the game, I had to try very hard to find it. So, it would be great if he could give a hint, feedback here. When transferring a picture from outside, I think it is a bad feature that the picture does not fully match the backdrop and there are color differences. Then it was very difficult to try to match the color…"

P4: "I think it would be more interesting if there were moving sprites. I think it looks a bit simplistic. I didn't like this look very…"

P8: "There was no feature that bothered me. I mean, there's nothing I can say I didn't like about that feature, frankly."

P13: "I think the worst feature was that I had to split the sound while adding it…"

P27: "The worst feature is the lack of props and sprites. Because I was going to make an animation about the teacher who lectures in the classroom. I looked, there is no classroom backdrop. There is no good teacher sprite either. I downloaded it to my computer from another web site and used it. I think it would be better if they were enriched. But this is not a very problematic situation, by the way."

P50: “The only thing I didn't like was the sounds the sprites made. Many of them weren't reflective of the sprite’s characteristics and I didn't like that they were very short-lived.”

Question 5. Can you explain why you like programming or not?

When the answers given to this question were examined, it was seen that 3 participants (P18, P26 and P51) stated that they did not like programming. The participants in question explained the reasons for their views by focusing on issues such as not finding programming fun, being irrelevant to the subject, and disliking the use of computers.

P18: “Programming is actually a fun job, but I didn't like it a bit. Because I've never been the kind of person involved in such things. I tried to do it as much as I could, but I can't say that it is an easy practice for me… Of course, I am an individual who is aware of how much things like software and code will be valuable in the future, and such applications inspire enthusiasm in children. And they can make children love the professions of the future. So although I personally dislike it, I cannot deny that it is needed.”
P26: “I didn't like it. Frankly, it's not fun and I didn't enjoy it visually."
P51: “When I was programming with Scratch, my ideas about programming changed a bit, but I can't say that I enjoyed it that much. So, I don't like using computers in general…”

When the views of 49 participants who stated that they enjoyed programming practices were examined, it was seen that 29 pre-service teachers stated that they got rid of their negative prejudices about programming, such as unsettling and scary. However, 12 participants stated that they have gained awareness of the importance of programming today, while 7 participants underlined that they discovered that programming is in all areas of life. 6 pre-service teachers emphasized that designing their own product with programming increased their self-confidence. 42 participants reported their satisfaction for their participation in this implementation process. Participants explained the reasons for their positive tendencies towards programming within the framework of the statements shown in the graph.

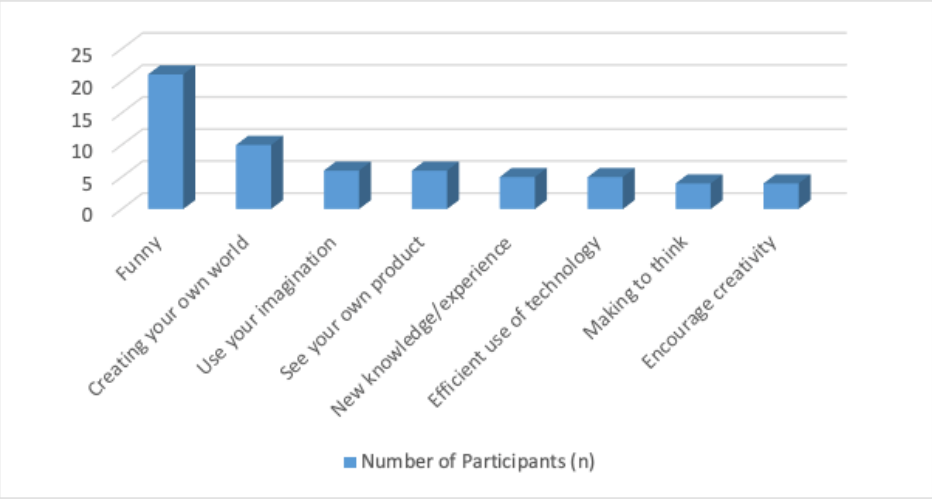

Figure 5. Reasons for Participants' Positive Views on Programming

The reasons why pre-service teachers like programming implementations: (1) It is fun; (2) to have the chance to create the worlds and projects they imagine with these applications; (3) programming allows them to use and develop their imaginations; (4) have the opportunity to see their products concretely; (5) enable them to acquire new and different experiences and knowledge; (6) enabling them to become productive users of technology; (7) constantly prompting thinking and (8) triggering creative thinking and creating an environment for this. Examples of participants' responses are presented below.

P1: "I was really interested in programming. I discovered that being able to do something digitally is a wonderful feeling. In general, I was very happy to see how the basic systems in the games I played were working."

P7: "It was a lot of fun. It makes us think all the time…"

P10: "I had a prejudice that programming was very difficult. I can say that the understandable structure of Scratch broke this prejudice. I even had more fun than I expected."

P17: "I enjoyed it very much. It was like I was just passing through technology before. Now I have entered the kitchen of animations and games."

P27: "I really enjoyed programming. It was an activity that I had never done until this year or even this class. By programming, you can build the worlds you want, imagine, and place the entity you want. It's all up to you. You are free. No limitation. What else?"

P35: "Programming is a very fun activity. Being able to see that you have created a project and to be able to share it in a concrete way adds self-confidence. I will definitely continue to produce projects in the field I want, both in the field of entertainment and education."

P45: "I enjoyed programming. I think programming improves people's thinking and designing skills. And it offers the opportunity to reflect it immediately on the screen. I can create my own thinking world… Something that is entirely my own feels beautiful. I also realized with this course that programming is in many parts of our lives. So, I feel lucky to have taken this class at school."

Question 6. Can you explain your views on the integration of programming practices into education?

When the answers given to this question were examined, it was seen that all participants, except for only one participant (P44), stated that it was necessary and important to integrate programming practices into education. P44, on the other hand, explained this view as follows:

P44: "It should not be used. My area includes primary school children. Primary school students take a lot of time to understand them, and this can tire the child. I prefer that he transforms his imagination by drawing a picture on paper, rather than transforming it into complex programming processes. This is more of an advanced level…"

51 pre-service teachers, who expressed positive views on the use of programming practices in educational processes, focused on various advantages of programming. At this point, the participants expressed their views within the framework of the outputs to be obtained because of the teacher's implementation of the designs developed by the teacher and the student's creation of their own designs. In line with these explanations, two categories as teacher designs and student designs were created. The category named teacher designs includes two sub-categories called impact on learner and impact on learning process. Student designs category includes two sub-categories as skill and perception.

Table 11. Participants' Views on the Effects of Programming Practices

Categories	Sub-categories	Codes	n
Teacher designs	Impact on learner	Increasing academic achievement	37
		Development of interest towards course	33
		Active participation to the course	21
		Increasing the sense of curiosity	18
		Development of interest towards teacher	18
	Impact on learning process	Increasing clarity	31
		Learning with fun	28
		Permanent learning	27
		Concretizing abstract issues	22
		Motivation development	18
		Learning by doing	16
		Reinforcing the subject matter	12
Learner designs	Skill	Creative thinking skills	42
		Problem solving skills	39
		Algorithmic thinking skills	29
		Group work skills	25
		Computer skills	19
		Analytical thinking skills	14
		Communication skills	10
		Critical thinking	5
		Motor skills	4
	Perception	Self-confidence development	28
		Efficient use of technology	21
		Innovative perspective	20
		Experience a sense of achievement	13
		Interdisciplinary work	11
		Establishing cause-effect relationships	7

Pre-service teachers stated that they could support their lessons with educational games, animations, or short stories of their own designs. In addition, some participants stated that they can find solutions with their own designs when they encounter problems such as not being able to find the content, they are looking for in digital materials that they can find readily available on the internet or that these materials have inappropriate content. However, they stated that programming practices are also included in the basis of being a producing society (n = 21) and raising technology pioneers (n = 20). The sub-category of influence on the learner, which is included in the category called teacher designs, focuses on the outputs that are thought to occur in students by supporting educational processes with digital designs. The most emphasized views of the category are that students' academic achievement in various fields such as science and mathematics will increase (n = 37) and interest in the course will increase (n = 33). When the effects of these designs on the learning processes are examined, the participant view, which states that the subjects that are especially difficult to understand can be conveyed more

easily with digital activities, and thus, it will be possible to increase the intelligibility of the subject (n =31), has been the most emphasized view in common. In addition, it was emphasized that games and animations would create a fun learning environment (n = 28), and it was stated that with these activities, students would be able to be in learning environments by doing and experiencing (n = 16). Most of the participants (n = 46) described programming practices as the need of the age and therefore underlined that children should be prepared for the future by taking this education at an early age. The individual skills sub-category in the category of student designs includes the participant responses that various skills such as creative thinking skills (n = 42) and problem-solving skills (n = 39) will develop of students who design their own projects with programming practices. However, the sub-category of individual perception in this category includes the various awareness expected to occur in students through the projects designed with programming and the positive changes in the way they perceive themselves. At this point, it is seen that individual changes such as programming and project creation processes will support the development of self-confidence (n = 28) and encourage the efficient use of technology by understanding (n = 21). Examples of participants' responses are presented below.

P4: "Students will understand the subjects better with enjoyable games. In addition, this way, students are prevented from getting bored in the lesson. The lesson becomes interesting. By teaching programming to students, it can be ensured that they develop their imaginations…"

P13: "It should definitely be used. When we become teachers, I think we will look for videos on the subject to visualize the subject. Sometimes we may not find these videos the way we want. Although the Internet offers many products, they may not be enough. There may be something objectionable in the videos. In such cases, for example, preparing our personalized animations will make our work easier and provide a better understanding of the subject…"

P37: "Programming drives innovation. It is a fundamental ability for students to lead technological advances. programming gives students the opportunity to be creative. They can do great projects. Programming builds confidence. It is a great strength for students to introduce the projects they have developed to their families and friends. programming brings success in other fields…"

P46: "Ensuring that children use their visual and auditory perceptions while learning helps them learn more permanently. In this respect, using these coded files while programming games or animations and teaching a subject to students will help them a lot. In this way, they will learn in a fun way…"

Question 7. Would you like to improve yourself in the field of programming? Why?

When the answers given to this question were examined, it was seen that two pre-service teachers (P17, P44) stated that they had no desire to improve themselves in the field of programming. The reasons for this situation were explained as difficult programming practices (P17) and interest in different fields (P44). One participant (P31) stated that he was not sure about this issue.

P31: “I'm not sure I want to develop more, but I would like to increase the knowledge I have now…”

P44: “No, I don't have such a thought right now. Why not in the future, but my priorities are mostly learning to play a musical instrument, things like sports…”

When the explanations of the participants (n = 49) who stated that they wanted to improve themselves in the field of programming were examined, the answers to the reasons for these requests formed the categories called individual development, individual interest, professional objective, professional perception, and digital age.

Table 12. The Reasons for the Participants' Desire to Develop in the Field of Programming

Categories	Codes	n
Individual development	Computer skills	31
	Creativity	28
	Collaboration skills	14
	Communication skills	9
Individual interest	Designing new projects	24
	Enjoyment	21
	Interesting	16
Professional objective	Funny course	42
	Permanent learning	30
	Facilitate understanding of subject matter	26
	Creative thinking students	23
	Attract students’ attention	15
	Effective education	10
	Mental development	5
	Giving a different perspective	4
Professional perception	Teachers applied to students’ needs	28
	Teachers teach while having fun	27
	Prestige	13
	Ease	9
Digital age	Modernization	34
	Necessity	32
	Financial gain	19

Within the framework of the individual development category, there are answers that focus on various developments that pre-service teachers think will occur in themselves with programming practices. At this point, it is seen that especially the developments related to computer skills and creativity levels are emphasized. The individual interest category includes the answers of the participants who stated that they wanted to design new projects at the end of the implementation process they participated in, and that they started to enjoy programming and algorithm creation studies or find these studies interesting. The category named professional objective focuses on the outputs that pre-service teachers plan to create by integrating programming practices into their teaching processes in their future professional lives. It has been seen that they believe that they will create fun and easy-to-understand lesson processes with programming and that they will achieve various goals such as ensuring permanent learning. The professional perception category includes the opinions of the participants who evaluate programming within the scope of the teaching profession. At this point, it has been stated that students have close relations with technology and therefore teachers need to use new technologies to speak the same language with their students and that knowing programming will bring prestige to the teacher.

Participant views based on the needs and conditions of the age in which they live formed the digital age category. Within the scope of this category, it was seen that pre-service teachers mentioned the importance and necessity of primary school students learning to code. Similarly, it was observed that there were participants who stated that they wanted to exist in the future by adapting to the conditions of the age and thought that they could provide financial gain by serving in various fields with programming. Examples of participants' responses are presented below.

P19: "I aim to improve myself in this field, produce more professional and quality content and teach while having fun as a teacher…"

P22: "… Technology is constantly changing; the world is changing. That's why you must keep up with it. For example, in this period, everyone is at home, students are learning with online education. Companies work from home with computer programs. That's why technology was necessary. I would also like to contribute to them. After all, we will be a classroom teacher and we must improve ourselves in every field to do this in the best way. We must also prepare our students for the future by developing them…"

P27: "Of course I would. I am the educator of the future. The more I improve myself until I start my profession, the better for both me and my students. For example, I will explain a topic. I prepare an animation; I open it on the smart board of the class. It grabs the attention of my students and stays in their minds. Or just finished a topic. I can prepare a game and play it with my students by drawing attention to some points that are aimed to be learned…"

P41: "I would definitely like to. Because the work you do is tangible, something tangible happens. I also want to produce something that will benefit someone…"

P50: "Yes, I would like to. Because after this course, after the games and animations I made, I felt very advanced in this direction as a person who does not use and cannot use computers much. Even looking at the little things I made myself, I realized how limitless I could make it, and that made me even more curious…"

CONCLUSION and DISCUSSION

The purpose of this study is to determine the views of pre-service elementary teachers about programming after the implementation process which they have experienced with Scratch. For this purpose, two structured interview processes as the pre-interview and the final interview were carried out. When the findings obtained during the pre-interview process were examined, it was seen that some participants reported that the programming practices included various letters, symbols or symbols that did not make sense to them. Similarly, participants who gave examples of programming languages also presented text-based programming environments as examples. At this point, it is noteworthy that the participants focused on text-based programming and none of the participants mentioned examples or explanations about block-based programming environments. In addition to this, when the examples given by the pre-service teachers for programming

environments were examined, the examples of Bitdegree, Codecademy, Coursera, edX, Khan Academy, Treehouse and Udemy were found remarkable. As a matter of fact, the examples in question are not described as programming languages. For example, Bitdegree is a platform that offers online courses, and it shares lecture videos on various subjects for a fee within the section called "coding and programming". Codecademy, an online training platform, provides training on programming languages such as Python and Java, and markup languages such as HTML. Similarly, Coursera, edX, Khan Academy, Treehouse, and Udemy are platforms that offer training in a wide variety of topics and fields. However, there is no programming language called C+. In addition, an event platform called Hour of Code is in active service, although there is no program called Code Hour. It was also seen that most of the participants did not have any experience with programming. However, it was remarkable that most of the participants who stated that they had experience did not remember details such as the name or use of the programming environment. It is thought that the effect of long-term experience on active technology use may be among the reasons for this situation.

When the findings obtained in the post-interview process are examined, it is a remarkable finding that the participants stated that they found the programming process with Scratch difficult at the beginning of the implementation process but realized that they were wrong with the progress of the process and increasing sample practices, and that they saw that Scratch offers an easy use for programming. While explaining the reasons for this situation, pre-service teachers mentioned the importance of practicing and that they started to gain familiarity by getting to know the programming language in this process. In this context, it can be stated that long-term first-hand experience has a positive effect on programming processes. As a matter of fact, pre-service teachers realized that the programming process, which they perceived as difficult or complex at the beginning of the implementation process, could be done easily. However, when the pre-service teachers' likes and dislikes about programming practices were examined, it was seen that most of the participants got rid of their prejudices and uneasiness towards programming and stated that they liked programming very much. However, only three participants stated that they did not like programming. At this point, it was remarkable that the participants, other than one participant, found programming fun and emphasized that they were aware of the importance of programming. As a matter of fact, at the end of the implementation process, it was determined that all participants, except one, developed a positive tendency and perception towards programming. This result is similar to the relevant literature. For example, in the study conducted by Fesakis and Serafeim (2009) with 35 university students, Scratch was introduced, and the participants developed their own designs. In the results of the research, it was stated that there was a positive effect on the views of future teachers about programming through Scratch. It is thought that this result obtained in this study is influenced by the features of Scratch to be described as user-friendly and the disappearance of prejudices towards programming through these features. In addition, all of the participants who stated that they did not like programming stated that these technologies should be integrated into education. At this point, one participant who stated that he liked coding stated that these implementations were advanced and explained that programming should not be used in educational processes.

According to the results of this research, the participants mostly started to show a positive tendency towards programming. In addition, most pre-service teachers reported their desire to improve themselves in this field and that they plan to use programming technologies in their future professional lives. It is thought that the fact that Scratch is based on block-based programming features is effective on this situation. Indeed, programming is intended to be attractive and accessible to children and non-majors through programming environments such as Scratch, which are proposed to support the development of skills in using programming languages (Fesakis & Serafeim, 2009). In addition to this, it can be stated that it is important to use block-based programming environments such as Scratch in teacher education so that children can enter programming at an early age. In this context, it should be underlined that visual programming languages not only allow programming to be more accessible to early age individuals, but also offer more opportunities for innovation and discovery (Alturayeif, Alturaief & Alhathloul, 2020). As a matter of fact, Scratch's features such as intuitive, motivating, and ideal for collaboration make this programming environment outstanding (Olabe, Olabe, Basogain & Castaño, 2011).

ETHICAL TEXT

In this article, journal writing rules, publishing principles, research and publishing ethics rules, journal ethics rules are followed. The author is responsible for all kinds of violations related to the article. Ethics committee approval was obtained for this study from Ordu University Social and Human Sciences Ethics Committee with the decision numbered 2021-128.

Author(s) Contribution Rate: The author's contribution to this article is 100%.

REFERENCES

Alturayeif, N., Alturaief, N., & Alhathloul, Z. (2020). DeepScratch: Scratch programming language extension for deep learning education. *(IJACSA) International Journal of Advanced Computer Science and Applications, 11*(7), 642-650.

Ashrafzadeh, A., & Sayadian, S. (2015). University instructors' concerns and perceptions of technology integration. *Computers in Human Behavior, 49*, 62-73. https://doi.org/10.1016/j.chb.2015.01.071

Attard, L., & Busuttil, L. (2020). Teacher perspectives on introducing programming constructs through coding mobile-based games to secondary school students. *Informatics in Education, 19*(4), 543-568. http://doi.org/10.15388/infedu.2020.24

Başarmak, U., & Hamutoğlu, N. B. (2019). Ortaokul öğrencilerinin "KOD Adı 2023" projesi eğitimine yönelik görüşleri. *Gazi Eğitim Bilimleri Dergisi, 5*, 55-66. http://doi.org/10.30855/gjes.2019.os.01.004

Bers, M. U. (2018). Coding and computational thinking in early childhood: The impact of ScratchJr in Europe. *European Journal of STEM Education, 3*(3), 08. https://doi.org/10.20897/ejsteme/3868

Chang, C. K. (2016, July 10-14). *Using Computational thinking patterns to scaffold program design in introductory programming course* [Paper presentation]. 2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI). Kumamoto, Japan. IEEE. http://doi.org/10.1109/IIAI-AAI.2016.27

Cohen, L., Manion, L., & Morrison, K. (2007). *Research methods in education*. Routledge.

Council of Higher Education. (2018). *Sınıf öğretmenliği lisans programı*. https://www.yok.gov.tr/Documents/Kurumsal/egitim_ogretim_dairesi/Yeni-Ogretmen-Yetistirme-Lisans-Programlari/Sinif_Ogretmenligi_Lisans_Programi09042019.pdf

Dolenc, K., & Aberšek, B. (2015). TECH8 intelligent and adaptive e-learning system: Integration into technology and science classrooms in lower secondary schools. *Computers & Education, 82*, 354-365. https://doi.org/10.1016/j.compedu.2014.12.010

Fessakis, 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. https://doi.org/10.1016/j.compedu.2012.11.016

Fesakis, G., & Serafeim, K. (2009). Influence of the familiarization with "Scratch" on future teachers' opinions and attitudes about programming and ICT in education. *ACM SIGCSE Bulletin, 41*(3), 258-262. https://doi.org/10.1145/1595496.1562957

Göncü, A., Çetin, İ., & Şendurur, P. (2020). Information Technology and Software Course Teachers' Opinions on Computing Education *Mersin University Journal of the Faculty of Education, 16*(2), 301-321. http://doi.org/10.17860/mersinefd.665725

Grover, S., & Basu, S. (2017, March 3-5). *Measuring student learning in introductory block-based programming: Examining misconceptions of loops, variables, and boolean logic* [Paper presentation]. 2017 ACM SIGCSE Technical Symposium on Computer Science Education. Washington, USA. https://doi.org/10.1145/3017680.3017723

Guba, E. G. (1981). Criteria for assessing the trustworthiness of naturalistic inquiries. *ECTJ, 29*(2), 75-91. https://doi.org/10.1007/BF02766777

Hainey, T., Baxter, G., & Ford, A. (2020). An evaluation of the introduction of games-based construction learning in upper primary education using a developed game codification scheme for scratch. *Journal of Applied Research in Higher Education, 12*(3), 377-402. https://doi.org/10.1108/JARHE-02-2018-0031

Lindberg, R. S., Laine, T. H., & Haaranen, L. (2019). Gamifying programming education in K-12: A review of programming curricula in seven countries and programming games. *British Journal of Educational Technology, 50*(4), 1979-1995. http://doi.org/10.1111/bjet.12685

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. https://doi.org/10.1145/1868358.1868363

MIT Media Lab. (2021). *Scratch*. http://scratch.mit.edu

Miles, M. B., & Huberman, A. M. (1984). Drawing valid meaning from qualitative data: Toward a shared craft. *Educational Researcher, 13*(5), 20-30. https://doi.org/10.3102/0013189X013005020

Mladenović, M., Boljat, I., & Žanko, Ž. (2018). Comparing loops misconceptions in block-based and text-based programming languages at the K-12 level. *Education and Information Technologies, 23*(4), 1483-1500. https://doi.org/10.1007/s10639-017-9673-3

Noeth, R. J., & Volkov, B. B. (2004). *Evaluating the effectiveness of technology in our schools, ACT Policy Report*. http://files.eric.ed.gov/fulltext/ED483855.pdf

Olabe, J. C., Olabe, M. A., Basogain, X., & Castaño, C. (2011). Programming and robotics with Scratch in primary education. In A. Méndez-Vilas (Ed.), *Education in a technological world: Communicating current and emerging research and technological efforts* (pp. 356-363). Formatex.

Omona, J. (2013). Sampling in qualitative research: Improving the quality of research outcomes in higher education. *Makerere Journal of Higher Education, 4*(2), 169-185. http://dx.doi.org/10.4314/majohe.v4i2.4

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B. & Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM, 52*(11), 60-67. https://doi.org/10.1145/1592761.1592779

Ruthmann, A., Heines, J. M., Greher, G. R., Laidler, P., & Saulters, C. (2010, March 10-13). *Teaching computational thinking through musical live coding in Scratch* [Paper presentation]. The 41st ACM Technical Symposium on Computer Science Education. New York, NY, United States. https://doi.org/10.1145/1734263.1734384

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. https://doi.org/10.1016/j.compedu.2016.03.003

Sheard, J., Simon, S., Hamilton, M., & Lönnberg, J. (2009, August 10-11). *Analysis of research into the teaching and learning of programming* [Paper presentation]. The Fifth International Workshop on Computing Education Research W*orkshop*. Berkeley, CA, USA. https://doi.org/10.1145/1584322.1584334

Stake, R. E. (1995). *The art of case study research*. Sage.

Suzuki, R., Kato, J., & Yatani, K. (2020). ClassCode: An Interactive Teaching and Learning Environment for Programming Education in Classrooms. *arXiv preprint arXiv:2001.08194*. http://arxiv.org/abs/2001.08194

Weintrop, D. (2019). Block-based programming in computer science education. *Communications of the ACM, 62*(8), 22-25. https://doi.org/10.1145/3341221

Weintrop, D., & Wilensky, U. (2017). Comparing block-based and text-based programming in high school computer science classrooms. *ACM Transactions on Computing Education (TOCE), 18*(1), 1-25. https://doi.org/10.1145/3089799

Yoshihara, K., & Watanabe, K. (2016, July). Practice of programming education using scratch and NanoBoardAG for high school students. In *2016 10th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS)* (pp. 567-568). IEEE.