Research Article

HOW DOES TEACHING PROGRAMMING THROUGH SCRATCH AFFECT PROBLEM-SOLVING SKILLS OF 5th AND 6th GRADE MIDDLE SCHOOL STUDENTS? [1]

Özkan VATANSEVER
Teacher, MEB, ozkanvatansever@hotmail.com
ORCID Number: 0000-0001-5949-9107

Şehnaz BALTACI GÖKTALAY
Assoc. Prof. Dr., Uludag University, sehnazbg@uludag.edu.tr
ORCID Number: 0000-0003-1961-1840

Received: 19.01.2018 *Accepted: 29.08.2018*

ABSTRACT

It is important to teach students programming at an early age so that they do not have difficulty in understanding the programming logic when they reach the age of undergraduate education. Scratch, a visual two-dimensional programming tool, has been developed to accomplish this purpose. There are studies in the literature related to Scratch on problem-solving skills, but they are not directed at the steps of problem solving. In order to find a proper solution to a problem, it is necessary to carry out each problem-solving step in an appropriate way. Based on this need, the aim of this study is to investigate how teaching programming through Scratch affects the problem-solving skills of 5th and 6th grade middle school students and to identify the steps that students go through to solve ill-structured problems. It is also intended that these steps guide teachers in terms of the pedagogies for teaching coding. This study utilized an explanatory design, one of the mixed method research designs. In the quantitative stage of the study, a one-group pretest-posttest model, one of the simple experimental models, was used. In the qualitative stage, a case study model was used as the research design. A total of 226 5th and 6th grade students, including 109 males and 117 females, constituted the sample of the study. The data were collected using a problem-solving skills scale and interview questions. A statistically significant difference was found between the mean scores of the first and second measurements of the problem-solving skills scale according to the Wilcoxon signed-rank test result. According to the results, when teaching programming through Scratch, it is recommended to have students design games, as an alternative method, for them to develop problem-solving skills. The solution of a problem is facilitated when students construct a game in their minds in accordance with real-life conditions and design and detail it in accordance with computational expressions.

Keywords: Scratch, problem-solving skills, teaching programming, coding, middle school students.

[1] This study was presented as an oral presentation at the “International Congress of Eurasian Social Sciences” in 6-8 April 2017. This study is produced from Özkan VATANSEVER's master thesis entitled " Examining The Effects Of Using Scratch Programming on 5th and 6th Graders' Problem Solving Skills" conducted under the consultancy of Assoc. Prof. Dr. Şehnaz BALTACI GÖKTALAY.

INTRODUCTION

The results of many national and international studies have shown that problem-solving skills are among the subjects where elementary and middle school students are least successful. In the 21st century, individuals have to be good at digital media literacy, innovativeness, creativity, critical thinking, problem solving and reflective thinking skills (Trilling and Fadel, 2009). In today's complex and rapidly developing world, we need young students who have better problem-solving skills than ever. The extent to which these skills are provided to students in the process from elementary school to university in Turkey is a matter of debate that continues to be relevant persistently.

It is important to teach students programming at an early age so that they do not have difficulty in understanding the programming logic when they reach the age of undergraduate education. Scratch, a visual two-dimensional programming tool, has been developed to accomplish such a purpose (Karabak and Güneş, 2013). Scratch is one of the most commonly used programming languages in today's programming education. Scratch is used in the Introduction to Computer Science course at Harvard University, one of the most respected universities in the world. Scratch keeps up with the era with its constantly updated and evolving structure (Resnick et al., 2009). The Scratch program is recommended as the first language for young learners to learn programming (Malan and Leitner, 2007). In the Scratch program, blocks are used to write codes. Since the block system is based on the drag and drop method, programs can be created quickly and easily. Programming a game in Scratch is similar to playing Lego. Blocks are combined as in the Lego game. If the right blocks are combined, a beautiful program, animation or game emerges.

The Information Technologies and Software course is a compulsory course for 5th and 6th grade students, with two course hours a week, and it is an optional course for 7th and 8th grade students, with two course hours a week. The subjects of the course determined in the curriculum published by the Ministry of National Education are as follows:

1. Informatics Literacy
2. Communicating, Sharing Information and Expressing Oneself by Using Information Technologies
3. Doing Research, Structuring Information and Working Collaboratively
4. Problem Solving, Programming and Developing Original Products
    - Can develop a strategy to solve a problem and complete a project and can use different perspectives and approaches when finding a solution.
    - Can recognize authoring and programming languages and can effectively use at least one authoring/programming language.
    - Can create models, simulations, and animations to examine systems and topics (Milli Eğitim Bakanlığı [MEB], 2012).

According to Jonassen and Kwon II (2001), even in the most contemporary learning environments, learning goals assume individuals to have problem-solving skills. They have classified the types of problems to be used in problem-based learning environments as "well-structured problems" and "ill-structured problems." The solution process of well-structured problems is known, and such problems are well-defined. Ill-structured problems have multiple solutions. Students constantly solve well-structured problems throughout their educational lives. What is important here is to create environments which will provide opportunities for solutions to ill-structured problems.

It is thought that the ill-structured problem environments can be created in Scratch. Ill-structured problems can be a bit difficult to solve because it is necessary to use information from multiple disciplines. Therefore, using the Scratch program will also support the learning of knowledge in different disciplines (such as mathematics, computer science, linguistics, and social studies) (Scratch, 2018). By using Scratch, students can design their own projects and apply creative solutions to problems they face in real life (Karabak and Güneş, 2013; Lee, 2011). The Ministry of National Education encourages the use of Scratch by posting training videos on how to use Scratch on their website http://scratch.eba.gov.tr. The use of the Scratch program is thought to be useful in the field of "Problem Solving, Programming and Developing Original Products." By using Scratch, students will solve ill-structured problems that they will encounter when designing games.

Kalelioğlu and Gülbahar (2014) have investigated how teaching programming through Scratch affects problem-solving skills of 5th grade students. They carried out an exploratory mixed method research study on 49 5th grade students. They taught a one-hour course per week for five weeks. They analyzed the problem-solving skills scale, which they used in their pretest-posttest design semi-experimental study, by running a paired samples t-test. According to the paired samples t-test results, there were no significant differences between the mean scores that the students received from the problem-solving skills scale. Brown et al. (2008) investigated the use of Scratch to teach problem-solving skills to middle school students. A total of four Scratch lessons were given to 113 middle school 5th and 6th grade students for 45 minutes per week during a month. The students in the experimental group had higher problem-solving skills than the control group students. Shin and Park (2014) administered a cognitive testing instrument for problem-solving capacities consisting of five factors to 46 6th grade students in order to measure the effect of programming in the Scratch environment on the problem-solving capacities of middle school students. In the study, mathematical logic problems were solved through Scratch programming. According to the results of the study, problem-solving skills of the students were positively affected. A statistically significant difference was found in the ability of thinking differently, decision making, and planning in favor of the experimental group. Nam, Kim and Lee (2010) examined how a course they prepared to teach Scratch programming to 60 6th grade students affected problem-solving skills. During a four-week period, eight hours of training was given on Scratch programming. The PISA 2003 problem-solving test was used as a pretest and posttest, modified to suit the students' level. A statistically significant difference was found between the problem-solving ability scores of the experimental group students and those of the control group students. Akcaoglu and Koehler (2014) examined the effect of

learning through game design on students' problem-solving skills. The scores of the experimental group was significantly higher than those of the control group in each of the following skills: system analysis and design, decision-making, and problem-solving.

When the studies in the relevant literature are examined, it can be seen that there are different results for the effect of programming with Scratch on problem-solving skills. In earlier studies, psychological scales, cognitive tests, and PISA tests were used. In our study, however, we examined the problem-solving steps. In order to achieve a healthy solution to a problem, it is necessary to carry out each problem-solving step in a healthy way. Our study reveals the steps students use in the problem-solving process in the Scratch environment. Moreover, it is intended that these steps guide teachers in terms of the pedagogies for teaching coding. To the best of our knowledge, we have not encountered a scale where problem-solving steps have been used to solve ill-structured problems in publications related to Scratch.

It is important to determine the level of problem-solving skills acquired as one of the basic skills in the curricula, which has been implemented since the 2005–2006 school year. It is thought that students design games when learning programming with Scratch, as an alternative method, to improve their problem-solving skills. The game design process naturally involves creating designs and solving problems. Using game designs to improve problem-solving skills can be considered an ideal match. The aim of this study is to investigate how teaching programming with Scratch affects the problem-solving skills of 5th and 6th grade middle school students and to identify the steps that students go through to solve ill-structured problems.

Research Questions

1. Do the problem-solving skills of students differ significantly when learning programming through Scratch?
   a. Do they differ significantly based on gender?
   b. Do they differ significantly based on grade level?
2. During the programming through Scratch, what are the steps that students go through to solve ill-structured problems when (a) interpreting and problem representation, (b) developing solutions and monitoring of solution processes, and (c) making justifications and evaluating their problem-solving processes?

METHOD

This study utilized an explanatory design, which is one of the mixed method research designs. In the quantitative stage of the study, a one-group pretest-posttest model was used, which is one of the simple experimental models. In the qualitative stage, a case study model was used as the research design. Creswell and Clark (2007) describe a mixed method research as an approach that allows to collect, analyze and integrate data using qualitative and quantitative research methods. The reason why the mixed method research is used

is because it is intended to reach more detailed data to be able to explain the underlying problematic situations. In this study, the information that was revealed through the first research question was examined in depth through the second research question. The study is a mixed study when it is considered as a whole. When carrying out an explanatory design, first, quantitative data are collected and analyzed, followed by the collection of qualitative data to complement the quantitative data (Büyüköztürk, Çakmak, Akgün, Karadeniz and Demirel, 2017). The most important limitation of the simple experimental approach is that it does not contain control groups. The case study is a research method based on the question of how and why, allowing the researcher to investigate a fact or phenomenon in depth. Information obtained through observations, interviews and documents can be analyzed using methods such as content analysis and descriptive analysis (Yıldırım and Şimşek, 2013).

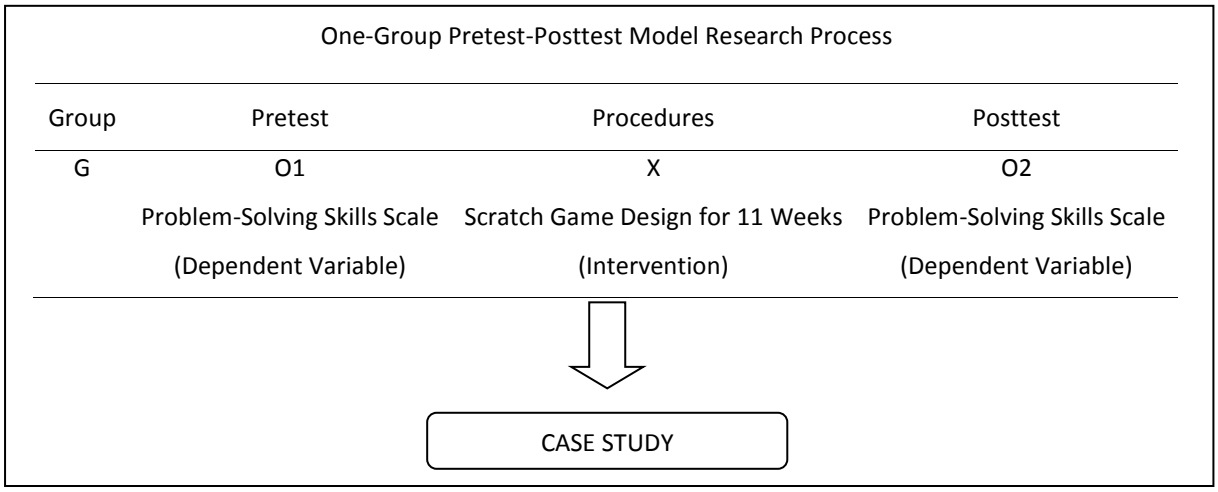
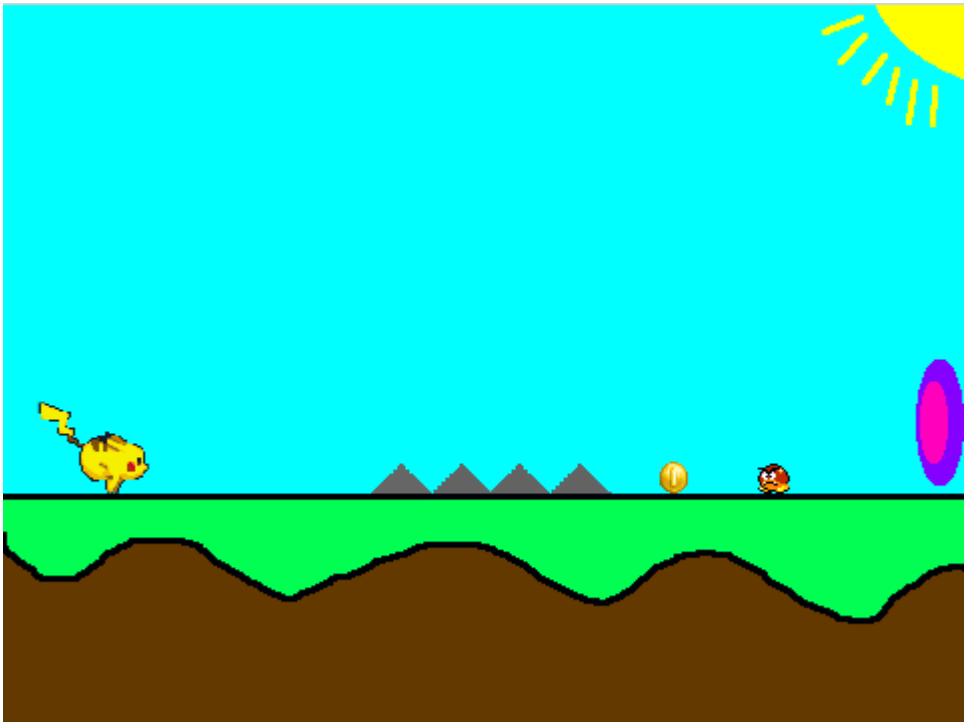
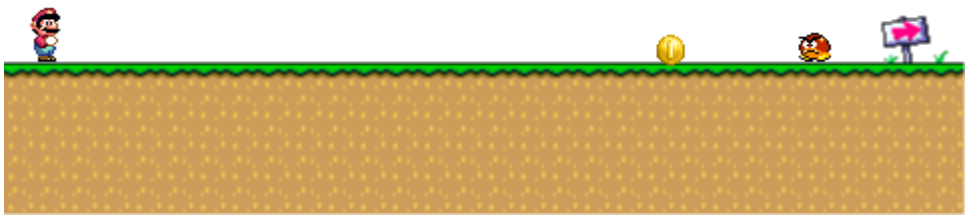


Figure 1. Explanatory Design Research Process



Picture 1. Pikachu Game Consisting of 7 Levels Designed with 5th Graders



Picture 2. Pikachu Game Consisting of 6 Levels Designed with 6th Graders

Data Collection Instruments

The Problem-Solving Skills scale — developed by Ge (2001) for solving ill-structured problems and translated into Turkish by Coşkun (2004) — was administered as a pretest and a posttest to determine the problem-solving skills of 5th grade and 6th grade students. During the case study, 27 students were asked certain

interview questions about the problem-solving steps that were derived from the scale to determine the procedures that they followed when solving ill-structured problems.

Table 1. Interview Questions on Problem-Solving Steps (Ge, 2001)

Problem-Solving Steps	Interview Questions
Interpreting and problem representation	What do you do before you start solving a difficult problem in Scratch?
Developing solutions and the monitoring of solution processes	What do you do when you are working on a problem in Scratch?
Making justifications and evaluating problem-solving processes	What do you do when you finish working on a problem in Scratch?

Sample

A total of 226 5th and 6th grade students, including 109 males and 117 females, constituted the sample of the study. The students were studying at a middle school in Bursa province. This study was carried out in 11 weeks for two course-hours per week in the 2014–2015 school year. The students used the Scratch program for the first time. The games in Picture 1 and Picture 2 were coded in the Scratch program.

The stratified purposeful sampling method is preferred in order to compare certain sub-groups of interest, and to illustrate and describe the characteristics of the groups. In this method, it is possible to work with subjects who can be accessed easily, and a fixed number of elements can be selected from all layers. In the criterion sampling, however, units that meet a specified criterion are sampled. It can be used when it is desired to perform an in-depth follow-up study after a quantitative study (Büyüköztürk et al., 2017).

The scale was used in a one-group pretest-posttest model, one of the simple experimental models for quantitative data collection. A total of 226 students were selected by using the stratified purposeful sampling method, one of the non-random sampling methods. The students were at the 5th and 6th grade, at which the Information Technologies and Software course is a compulsory course. The scale was administered to these students. During the case study, one of the qualitative research methods, the criterion sampling method, which is a non-random purposeful sampling method, was used to select students. These students had the potential to offer rich data from a knowledge point of view. They were highly capable of self-expression, and their grade point averages were 85 or above. In-depth data were collected from 27 students using the interview method.

Data Analysis

Analysis of the Quantitative Data

The problem-solving skills scale was administered as a pilot study to 100 students who were at the 7th grade. The reliability coefficient was found to be .83. Quantitative data, which were collected in two steps as the pretest and posttest, were subjected to the Kolmogorov-Smirnov test. It was found that the data were not

normally distributed according to the results of this test. The Skewness value was 1.814 and the Kurtosis value was 4.536. In line with this, it was decided to perform statistical analyzes using non-parametric tests. The Wilcoxon signed-rank test, one of the nonparametric tests, was used to determine whether there was a significant difference between the students' mean scores of problem-solving skills. The Mann-Whitney U, which is also a nonparametric test, was used to determine whether there was a significant difference between the students' mean scores of problem-solving skills according to gender and grade level.

Analysis of the Qualitative Data

Since the conceptual framework of the study was pre-determined based on a literature review and the quantitative study, it was preferred to analyze the qualitative data using the descriptive research method, which is the most appropriate method for such situations (Yıldırım and Şimşek, 2013). The analysis of the data takes place in four stages: (1) coding data, (2) finding themes, (3) arranging the codes and the themes, and (4) identifying and interpreting the findings (Yıldırım and Şimşek, 2013). Within the scope of the study, these four steps were carried out as follows:

1. **Coding Data:** An Excel table was created based on the interview data to include the participants' pseudo names in the first column, the answers they gave to the questions in the second column, and the relevant codes in the third column. Since the themes used in the descriptive analysis were taken from the problem-solving literature, there was already an existing structure for the analysis of the data, and this theoretical framework facilitated the analysis of the data. The new codes that emerged during the coding according to this theoretical framework were also included in the list. The information about the randomly selected codes from the data and the codes where the answers of the participants fit into are as shown in Table 2.

Table 2. Descriptive Analysis: A Sample Coding

Pseudo Participant Names	Participants' Answers to Questions	Code
Kardelen	I design the project in my mind, I plan what I am going to do, I construct a scenario, and I think about what the puppet will do.	Construction
Ahmet	I think about what the game might be like. I imagine it in my head. I construct it using the Scratch characters. I connect it with real life.	Construction
Büşra	I think about what code blocks I can use to do it. I draw the scenario on paper.	Design

2. **Finding Themes:** Yıldırım and Şimşek (2013) defines "categories or themes" as the categorization and aggregation of codes or concepts under a more general title. If too many themes emerge, a more general theme can emerge, covering the earlier themes. The themes created during the descriptive analysis for the second research question were organized according to the problem-solving steps of Ge (2001), and they were collected under 3 headings: (1) interpreting and problem representation, (2) developing

solutions and the monitoring of solution processes, and (3) making justifications and evaluating problem-solving processes.

3. **Organizing Codes and Themes:** During the analysis of the second research question, 9 codes were derived under 3 headings in total. When performing this analysis, we ensured that the codes and themes were simple and understandable, that codes explained the idea units, and that the themes could aggregate codes under general headings.
4. **Identification and interpretation of findings:** A descriptive analysis was carried out at this stage. According to Yıldırım and Şimşek (2013), it is necessary for the researcher to describe the data obtained at this last stage by codes and themes, and then clarify the connection between the data and the research problems by presenting quotes as well as by exemplifying, explaining, interpreting and visualizing the data. Throughout this study, each student was given pseudo names while presenting quotes from the students.

FINDINGS (RESULTS)

Does teaching programming using Scratch significantly affect the problem-solving skills of students?

The data to be compared mean-wise were in the same group and were obtained through two consecutive measurements. At our disposal, there were pairs of data generated by test results from the same students at different times. The nonparametric test to determine whether there is a statistically significant difference between the averages of the data values obtained after two consecutive measurements on the same data source is called the Wilcoxon signed-rank test.

Table 3. Normality Test

	Kolmogorov-Smirnov[a]			Shapiro-Wilk		
	Statistics	N	p	Statistics	N	p
difference	.092	226	.000	.967	226	.000

Since the result of the Kolmogorov-Smirnov test for normality was p<.05, the array of differences was not normally distributed. The Wilcoxon Signed-Rank Test, a non-parametric test, was administered to determine whether there was a significant difference between the students' scores before and after the application. The test results are shown in Table 4 and Table 5 below.

Table 4. Wilcoxon Signed-Ranks Test: Descriptive Statistics of the Rank of Difference Scores

		N	Mean Rank	Sum of Ranks
posttest - pretest	Negative Ranks	66a	92.77	6123.00
	Positive Ranks	147b	113.39	16668.00
	Ties	13c		
	Total	226		

a. posttest < pretest

b. posttest > pretest

c. posttest = pretest

According to the descriptive statistics of the rank of difference scores, the posttest scores of 66 students were lower than their pretest scores, the posttest scores of 147 students were higher than their pretest scores, and the pretest and posttest scores of 13 students were equal.

Table 5. Wilcoxon Signed-Ranks Test: The Significance Test Result

	posttest - pretest
Z	-5.859b
p	.000

According to the results of Wilcoxon signed-rank test in Table 5, there was a statistically significant difference between the students' pretest and posttest scores (Z=-5.859, p<.005). The fact that the difference scores were in favor of the positive ranks suggests that teaching programming with Scratch had a significant effect on the problem-solving skills of the students. The effect size (d=.389) value shows us that teaching programming through Scratch had a moderate effect on the problem-solving skills of 5th and 6th grade students in middle school.

When learning programming through Scratch, do the problem-solving skills of students differ significantly according to gender?

Table 6. Mann-Whitney U Test: The Significance Test Result According to Gender

	difference
Mann-Whitney U	6031.500
Wilcoxon W	12026.500
Z	-.703
p	.482

a. Group Variable: Gender

The result of the significance test of the Mann-Whitney U test, which tested the difference between the mean scores of male and female students, was found to be p=.482>.05. That is, when learning programming through Scratch, the problem-solving skills of the students did not differ significantly according to gender.

When learning programming using Scratch, do the problem-solving skills of students differ significantly according to grade level?

Table 7. Mann-Whitney U Test: The Significance Test Result According to Grade Level

	difference
Mann-Whitney U	5476.000
Wilcoxon W	12857.000
Z	-1.789
p	.074

a. Group Variable: Grade

The result of the significance test of the Mann-Whitney U test, which tested the difference between the mean scores of 5th and 6th grade students, was found to be p=.074>.05. That is, when learning programming using Scratch, the problem-solving skills of the students did not differ significantly according to grade level.

What are the steps that students go through in solving ill-structured problems when learning programming through Scratch?

The question "What do you do before you begin to solve a difficult problem?" — representing the items 1–5 in the scale that we used to measure problem-solving skills of the students — was related to the stage of interpreting and problem representation. The question "What do you do when working on the problem?" — representing the items 6–10 — was related to the stage of developing solutions and the monitoring of solution processes. The question "What do you do when you finish working on the problem?" — representing the items 11–15 — was related to the stage of making justifications and evaluating problem-solving processes.

What are the steps students go through during the stage of "interpreting and problem representation"?

The first interview question was asked to find out what steps the students went through during the stage of interpreting and problem representation. The frequency of these steps is given below.

Table 8. Interpreting and Problem Representation

Problem-Solving Steps	Frequency (N=27)
Construction	20
Design	7
Other Opinions	2

Construction

According to the frequencies in Table 8, it was found that the students (n=20) constructed the game in their minds by establishing relations with real life. Below are some of the student views.

Duygu said that "I design the game in my head and edit it. If I have to give an example, it would be like how we jump up." She stated that she constructed the action steps in the game with references to real life in her mind. Ahmet said that "I think about what the game might be like. I imagine it in my head. I construct it using the

Scratch puppets. I establish connections with real life." He endorsed the words of Duygu. He stated that he constructed the game by using real life as a reference, and by using Scratch puppets in his mind. Zülal said, "I focus on the problem. I play the game a few times. Ideas come to my mind when I play the game. I remember the information that is necessary at that moment. I analyze the problem a few times while playing the game. I design the sequence of the code blocks." She designed the sequence of the code blocks by constructing the game in her mind.

Burçak stated that he would get a hard time if he began combining blocks of code before constructing the game in his mind. He expressed the following words: "If I directly begin to work on code blocks without thinking, I get confused, and it becomes hard to do. I collect different solutions in my head. I consider what I know about this topic in my head. I design what I have to do for the problem." Kardelen stated that she designed the game according to the action of the puppets in her mind, by expressing "I design the project in my mind, I plan what I am going to do, I construct a scenario, I think about what the puppet will do."

İbrahim said, "I think about what I am going to do in my head. For example, if the subject is jumping, I consider whether x or y is going to increase when it jumps." He matched the game with mathematical expressions while constructing it in his mind. Scratch also helps to learn mathematical expressions. Muhammet confirmed the words of İbrahim with his discourse, "I design the game according to the characteristics of puppets in my mind. For example, I think about how the jump is going to be." As an example, the Mario puppet should jump when the arrow key is pressed. He needs to think about at which axis its value will change over time.

Design

According to the frequencies in Table 8, it is seen that the students (n=7) wrote in the Scratch environment the design of the game which they had constructed in their minds according to the chronological order of events. Below are some of the student views.

Emir stated that he put down his design in the Scratch environment by adding comments to the add comments section. He expressed that "I design the game in the add comment section of Scratch." Nisanur said that "I sort out, in bullet points, what I have to do to solve the problem. We cannot know which blocks of code to use without designing the game." She stated that she put down in the Scratch environment the action steps appropriate for the design according to the sequence of actions. Beyza said, "I write the scenario according to the code blocks into the add comments section." She put down the design of the game in the Scratch environment based on the code blocks.

Melike said, "First, I think about how I can do it in my imagination. I plan it and draw it. I design the game on my notepad." She stated that the Scratch program encouraged her to use her visual intelligence. With her words "I think about what code blocks I can use to do it. I draw the scenario on paper," Büşra also confirmed that she supported Melike's opinion.

What are the steps students go through during the stage of "developing solutions and the monitoring of solution processes"?

The second interview question was asked to find out what steps the students went through during the stage of developing solutions and the monitoring of solution processes. According to the frequencies in Table 9, the students (n = 26) stated that they referred to the design of the game while placing the code blocks.

Table 9. Developing Solutions and the Monitoring of Solution Processes

Problem-Solving Steps	Frequency (N=27)
Arranging code blocks according to design	26
Other Opinions	1

Arranging code blocks according to design

As can be understood from the following statements, it was necessary for students to use design-oriented approaches to solve ill-structured problems.

Duygu's words, "I place the code blocks according to the design of the game that I visualize in my head." supports Sıla's words, "I place the code blocks according to the order of actions in the scene." Ahmet agrees with the opinions of Duygu and Sıla, by expressing, "I determine the code blocks according to the actions of the puppets. I place the code blocks according to the order of the game." Muhammad, with his words, drew attention to the order in which actions take place: "I place the code blocs according to the order of design. I place appropriate code blocks for actions that will take place step by step."

Burçak said, "I examine code blocks and choose the ones that will work for me. I choose the initial code block. First, I choose which code block to add from the control menu. I finally arrange the order of the code blocks according to the design by dragging and dropping other code blocks onto the scene." He, thus, drew attention to the importance of code blocks corresponding to computational expressions in the control menu. He also sorted the code blocks according to the design. Emir said, "I place the code blocks according to the design that I write in the comments section." He set the order of the code blocks according to the mental design process, which he had put down in the Scratch environment.

What are the steps students go through during the stage of "making justifications and evaluating their problem-solving processes"?

The third interview question was asked to find out what steps the students went through during the stage of making justifications and evaluating their problem-solving processes. According to the frequencies in Table 10, all of the students stated that they run the program and moved on to the next stage if they did not receive any errors. They stated that, in case of error, they controlled the order of placement of the code blocks in the

corresponding section according to the design of the game. However, three students stated that they were checking the design of the game to find the error.

Table 10. Making Justifications and Evaluating Problem-Solving Processes

Problem-Solving Steps	Frequency (N=27)
Running the program and moving to the next step	27
Checking the code blocks according to the design	20
Checking the design	3
Other Opinions	2

CONCLUSION and DISCUSSION

This study was carried out to investigate students' problem-solving skills through their problem-solving steps. After teaching programming through Scratch, it was concluded that the problem-solving skills of the students were positively affected similar to the studies in the literature (Akcaoglu and Koehler, 2014; Brown et al., 2008; Calder, 2010; Nam et al., 2010; Shin and Park, 2014). Moreover, the problem-solving skills did not vary according to grade level and gender. According to the results of Kalelioğlu and Gülbahar (2014), there were no significant differences between the mean scores of the students' problem-solving skills. They stated that the lack of a significant difference in their research results may be due to the short training period and small sample size. As stated in the study of Karabak and Güneş (2013), the amount of time allocated for the field of "Problem Solving, Programming and Developing Original Products" in the annual plan of the curriculum of the Information Technologies and Software course is equivalent to 22 course hours for 11 weeks. The reason why there was a significant difference in our study may be that the students received 22 course hours of education, and the sample size was sufficient.

According to Robertson (2012), compared to playing a game, the process of designing a game is a cognitive process involving complex design tasks. Design tasks require the use of many interrelated variables and parameters to create a complete and functional system. The students had to use conditional expressions and loops (computational expressions) during the game design process in the Scratch program. Therefore, it was found in our study that game design projects force students to learn programming expressions similar to the previous studies (Adams, 2010; Adams and Webster, 2012; Claypool, 2013).

Considering the stage of interpreting and problem representation, the students constructed their games in their minds by referring to real life and put down in the Scratch environment the design of their games which they had constructed. The students stated that they would be challenged if they directly began combining code blocks before constructing the game in their minds. Gomes and Mendes (2007) explored the difficulties of learning programming and the solutions to those difficulties. They emphasized that many times students go through the process of solving a problem without fully understanding the problem. In the Scratch environment,

students define the problem by first constructing ill-structured design tasks of their games in their minds and then by putting it down in the Scratch environment. It is found that, students also learn mathematical expressions such as x and y axes, similar to previous studies (Calder, 2010; Lee, 2011; Resnick et al., 2009). In our study, it was also reported that, this act of students — constructing the game in their minds in accordance with real life conditions —supports the development of their imaginations, similar findings were found in the literature (Burke and Kafai, 2010; Giannakos, Jaccheri and Proto, 2013; Wilson and Moffat, 2010).

As understood from the student views, during the process of programming a game in Scratch, the stage of defining the problem is an important stage in solving ill-structured problems because it facilitates the solution process of the problem. The definition of the problem depends on the interpretation or understanding of the problem (Chi and Glaser, 1983; Jonassen, 1997; Voss and Post, 1988). When the problem is characterized well (Michaelson, 2015), it becomes relatively easier to devise a solution in the Scratch environment. In order to test possible solutions and to code information, students define constraints by detailing the initial state of the problem by using a design-oriented approach.

Students place code blocks according to the design of the game in the stage of developing solutions and the monitoring of solution processes. There is a design that students take as a reference. The game design of each student for the solution of the problem is different. As can be seen from the students' views, there is no single solution to ill-structured problems that can be determined by going through a specific decision-making process, and this finding is also consistent with the literature (Chi and Glaser, 1983; Jonassen, 1997). During the game programming process through Scratch, each student learns at his or her own pace (Gomes and Mendes, 2007) and gradually builds his or her own knowledge. Students also noted the importance of code blocks corresponding to computational expressions in the control menu at this stage. When we look at the relevant literature (Adams and Webster, 2012; Brown et al., 2008; Calder, 2010; Claypool, 2013), we see that the Scratch program is successful in teaching computational concepts. Game design tasks improve intrinsic motivation in learning computational concepts.

Considering the opinions of the students regarding the decision-making and evaluation stage, the majority of the students were designing their games correctly, and mistakes were mostly due to their inability to combine code blocks according to the design. The reason for this is that they did not design their games in accordance with the computational statements while detailing their designs. Accordingly, they had difficulty in finding the expressions corresponding to logical and conditional expressions, i.e., computational expressions, in the control menu of the Scratch program. The process of solving ill-structured problems can be considered as a design process. Students need to design their games in accordance with the computational expressions in the control menu. This process can be defined as a computational thinking process (Aho, 2012; Barr, Harrison and Conery, 2011; Michaelson, 2015; Wing, 2006).

Solving problems that are not conceptually structured can be thought of as a design process, not a systematic study for problem solving. The solution of problems is facilitated when students construct a game in their

minds in accordance with real-life conditions and design and detail it in accordance with computational expressions. When the students designed a game with the help of Scratch, they mainly performed the following actions in the order given: (1) They constructed the game in their minds in accordance with real life conditions, (2) put down in the Scratch environment the design of the game which they had constructed, (3) placed the code blocks according to the design, (4) run the program and go to the next stage, and (5) checked the code blocks in relevant sections according to the design in case of error. The students checked the relevant section according to their designs in the event of an error. This shows the importance of designing games according to computational expressions (Aho, 2012; Barr et al., 2011; Michaelson, 2015; Wing, 2006).

As can be understood from the students' views, each student has a different computational thinking process. It is possible to say that today it is necessary for individuals to be aware of different ways of thinking and to take decisions by using a computer logic when solving problems; that is to say, they need to adopt a computational thinking approach (Barr et al., 2011; Wing, 2006). The participants thought that the Scratch software used to teach programming can contribute to multifaceted thinking and creating alternative solutions. This view of the participants is important in that it suggests that the Scratch software may support computational thinking in these contexts.

SUGGESTIONS

According to the results, when teaching programming with Scratch, it is recommended to have students design games, as an alternative method, for them to develop problem-solving skills. When we examine the problem-solving steps of the students, it is recommended to implement a design-oriented approach for them to solve ill-structured problems in the Scratch environment and to successfully complete game projects. In order to provide guidance to students during the stage of interpreting and problem representation, animating puppets in the classroom environment together with the students with reference to real life will accelerate and facilitate the mental processes of students. Teachers are advised to guide students in detailing their game designs in accordance with the computational statements in the control menu of Scratch.

Although there has been a steady increase in the publications related to programming in the relevant literature since 2012, these publications do not focus on the pedagogies for teaching coding. The steps that students follow in the game programming process through Scratch can guide teachers in terms of the pedagogies for teaching coding. As the Scratch program is a two-dimensional visual programming tool, it is suggested to implement the study using the Alice program, which is a three-dimensional visual programming tool. In addition, it is recommended that the study is carried out using true experimental designs.

Acknowledgement: This research is funded by Bursa Uludag University Research Fund No: KUAP(E)-2013/32.

SCRATCH İLE PROGRAMLAMA ÖĞRETMEK 5. VE 6. SINIF ÖĞRENCİLERİN PROBLEM ÇÖZME BECERİLERİNİ NASIL ETKİLER?

TÜRKÇE GENİŞ ÖZET

GİRİŞ

Pek çok ulusal ve uluslararası araştırmaların sonuçları ilköğretim ve ortaöğretim öğrencilerinin en başarısız oldukları alanlar içerisinde problem çözme becerisinin olduğunu göstermiştir. 21. yüzyılda insanların sahip olması gereken beceriler dijital medya okuryazarlığı, yenilik getirme, yaratıcılık, eleştirel düşünme, problem çözme ve yansıtıcı düşünme becerileridir (Trilling ve Fadel, 2009). Bugünün karmaşık ve hızla gelişen dünyasında her zamankinden daha fazla problem çözme becerilerine sahip genç öğrencilere ihtiyacımız vardır. Türkiye'de ilkokuldan üniversiteye kadar olan süreçte, öğrencilere bu becerilerin ne derece kazandırıldığı ise sürekli güncelliğini koruyan bir tartışma konusudur.

Öğrencilerin lisans seviyesine geldiklerinde programlama mantığını kavramakta güçlük çekmemeleri için erken yaşlarda programlama eğitimi verilmesi önem taşımaktadır. Bunun için görsel iki boyutlu bir programlama aracı olan Scratch geliştirilmiştir (Karabak ve Güneş, 2013). Scratch, günümüzde programlama eğitiminde en çok kullanılan programlama dillerinden biridir. Dünyanın en saygın üniversitelerinden biri olan Harvard'da bilgisayar bilimlerine giriş dersinde Scratch kullanılmaktadır. Scratch, sürekli güncellenen ve gelişen yapısı ile de çağa ayak uydurmaktadır (Resnick vd., 2009). Genç öğrencilerin programlamayı öğrenmeleri için ilk dil olarak Scratch'i kullanmaları tavsiye edilmektedir (Malan ve Leitner, 2007).

İlgili alanyazındaki çalışmalar incelendiğinde Scratch ile programlama öğretiminin problem çözme becerisine etkisine yönelik farklı sonuçlar görülmektedir. Çalışmalarda; psikolojik ölçekler, bilişsel testler ve PISA testleri kullanılmıştır. Bizim araştırmamızda ise problem çözme adımlarına bakılmaktadır. Bilgimiz dahilinde, Scratch ile ilgili yayınlarda yapılandırılmamış problemlerin çözümüne yönelik problem çözme adımlarının kullanıldığı bir ölçek bulunmamaktadır. Problemin çözümüne sağlıklı bir şekilde ulaşabilmek için problem çözme adımlarının her birinin sağlıklı bir şekilde gerçekleştirilmesi gerekmektedir. Bizim çalışmamız öğrencilerin Scratch ortamında problem çözme sürecinde kullandıkları adımları ortaya çıkarmaktadır. Ayrıca, bu adımların kodlama öğretiminin pedagojisi açısından öğretmenlere rehberlik etmesi amaçlanmaktadır.

Problem çözme becerilerine sahip öğrenciler yetiştirmek için alternatif bir metot olarak Scratch ile programlama öğretiminde oyun tasarlanması düşünülmektedir. Oyun tasarımı süreci doğal olarak tasarım ve problem çözmeyi gerektirmektedir. Problem çözme becerilerini geliştirmek için oyun tasarlamak ideal bir eşleşme olarak kabul edilebilir. Bu çalışmanın amacı, Scratch ile programlama öğretiminin ortaokul 5. ve 6. sınıf öğrencilerinin problem çözme becerileri üzerindeki etkisini araştırmak ve öğrencilerin yapılandırılmamış problemleri çözerken uyguladıkları işlem adımlarını belirlemektir.

Vatansever, Ö. and Baltacı Göktalay, Ş. (2018). How Does Teaching Programming through Scratch Affect Problem-solving Skills of 5th and 6th Grade Middle School Students?, International Journal of Eurasia Social Sciences, Vol: 9, Issue: 33, pp. (1778-1801).

Araştırma Soruları

1. Scratch ile programlama öğretiminde öğrencilerin problem çözme becerileri anlamlı bir farklılık göstermekte midir?
   c. Cinsiyete göre anlamlı bir farklılık göstermekte midir?
   d. Sınıf seviyesine göre anlamlı bir farklılık göstermekte midir?
2. Scratch ile programlama öğretiminde öğrencilerin yapılandırılmamış problemleri çözerken; yorumlama ve problemin tanımlanması, çözümler geliştirme ve çözüm sürecini izleme, karar verme ve çözüm sürecini değerlendirme aşamalarında uyguladıkları işlem adımları nelerdir?

YÖNTEM

Karma yöntem araştırmalarından açıklayıcı desen kullanılan bu araştırmanın nicel aşamasında basit deneysel modellerden tek grup öntest / sontest model, nitel aşamasında ise araştırma deseni olarak durum çalışması kullanılmıştır. Karma araştırma yönteminin kullanılması problem durumlarını açıklamada daha detaylı verilere ulaşılmak istenmesinden kaynaklanmaktadır. Bu araştırmada birinci araştırma sorusuyla ortaya çıkartılan bilgiler ikinci araştırma sorusuyla derinlemesine incelenmiştir. Çalışma bir bütün olarak ele alındığında karma bir çalışmadır.

Veri Toplama Araçları

Yapılandırılmamış problemlerin çözümüne yönelik Ge (2001) tarafından geliştirilen, Coşkun (2004) tarafından Türkçe'ye çevrilen ölçek 5. ve 6. sınıf öğrencilerinin problem çözme becerilerini belirlemek amacıyla öntest ve sontest olarak uygulanmıştır. Durum çalışmasında, öğrencilerin yapılandırılmamış problemleri çözerken uyguladıkları işlem adımlarını belirlemek için, 27 öğrenciye ölçekteki problem çözme adımlarına ilişkin mülakat soruları yöneltilmiştir.

Tablo 1. Problem Çözme Adımlarına İlişkin Mülakat Soruları Ge (2001)

Problem Çözme Adımları	Mülakat Soruları
Yorumlama ve problemi tanımlama	Scratch'ta zor bir problemi çözmeye başlamadan önce ne yaparsın?
Çözümler geliştirme ve çözüm sürecini izleme	Scratch'ta problem üzerinde çalışırken ne yaparsın?
Karar verme ve problem çözme sürecini değerlendirme	Scratch'ta problem üzerinde çalışmayı bitirdikten sonra ne yaparsın?

Araştırma Grubu

Araştırmanın örneklemini Bursa ili Osmangazi ilçesindeki bir ortaokulda öğrenim gören 109 erkek, 117 kız öğrenci olmak üzere toplam 226 5. ve 6. sınıf öğrencisi oluşturmaktadır. Bu araştırma, 2014-2015 eğitim

öğretim yılında haftada iki ders saati olmak üzere 11 haftada gerçekleştirilmiştir. Öğrenciler Scratch programını ilk kez kullanmışlardır.

Verilerin Analizi

Nicel Verilerin Analizi

Problem çözme becerisi ölçeği, pilot çalışma olarak 7. sınıfa giden 100 öğrenciye uygulanmış ve yapılan çalışmada güvenirlik katsayısı 0,83 olarak bulunmuştur. Ön test ve son test olmak üzere iki aşamada toplanan nicel veriler Kolmogorov-Smirnov testine tabi tutulmuş ve bu test sonucuna göre normal dağılmadıkları tespit edilmiştir. Skewness değeri 1,814 ve Kurtosis değeri 4,536 olarak bulunmuştur. Bunun doğrultusunda parametrik olmayan testlerle istatistiksel analizlerin yapılmasına karar verilmiştir. Parametrik olmayan testlerden Wilcoxon işaretli sıralar testi ile öğrencilerin problem çözme becerileri puan ortalamaları arasında anlamlı bir farklılık olup olmadığına bakılmıştır. Parametrik olmayan testlerden Mann-Whitney U testi ile öğrencilerin problem çözme becerileri puan ortalamaları arasında cinsiyete ve sınıf seviyesine göre anlamlı bir farklılık olup olmadığına bakılmıştır.

Nitel Verilerin Analizi

Araştırmanın kavramsal çerçevesi, alanyazın tarama kısmında ve nicel araştırma kapsamında önceden belirlendiğinden, araştırmanın nitel kısmını gerçekleştirebilmek amacıyla, bu tür durumlar için en uygun yöntem olan betimsel araştırma yöntemi (Yıldırım ve Şimşek, 2013) kullanılarak verilerin analiz edilmesi tercih edilmiştir. Bu doğrultuda, verilerin analizi dört aşamada gerçekleştirilmiştir: (1) verilerin kodlanması, (2) temaların bulunması, (3) kodların ve temaların düzenlenmesi, (4) bulguların tanımlanması ve yorumlanması olmaktadır.

BULGULAR

Scratch ile programlama öğretiminin öğrencilerin problem çözme becerileri üzerinde anlamlı bir etkisi var mıdır?

Yapılan Kolmogorov-Smirnov testine göre verilerin normal dağılmadığı tespit edilmiştir ($p<0,05$). Bu nedenle, öğrencilerin uygulama öncesi puanları ile uygulama sonrası puanları arasında anlamlı bir fark olup olmadığını belirlemek için, parametrik olmayan bir test olan Wilcoxon İşaretli Sıralar Testi uygulanmıştır. Wilcoxon işaretli sıralar testinin sonucuna göre; öğrencilerin ön test ve son test puanları arasında istatistiksel olarak anlamlı bir fark gözlenmiştir ($Z=-5.859$, $p<.005$). Fark puanlarının pozitif sıralar lehine olması, Scratch ile yapılan programlama öğretiminin öğrencilerin problem çözme becerileri üzerinde anlamlı bir etkisinin olduğunu göstermektedir. Hesaplanan etki büyüklüğü ($d=0,389$) değeri; Scratch ile programlama öğretiminin ortaokul 5. ve 6. sınıf öğrencilerinin problem çözme becerileri üzerinde orta düzeyde bir etkisi olduğunu göstermektedir. Farklar puan dizisinin betimsel istatistiklerine göre; 66 öğrencinin son test puanı ön test puanına göre düşük

çıkmıştır, 147 öğrencinin son test puanı ön test puanına göre yüksek çıkmıştır, 13 öğrencinin ise ön test ve son test puanları eşit çıkmıştır.

Kız ve erkek öğrencilerin puan ortalamaları arasında fark olup olmadığı Mann-Whitney U testi ile sınanmıştır. Buna göre, Scratch ile programlama öğretiminde öğrencilerin problem çözme becerileri cinsiyete göre anlamlı bir farklılık göstermemektedir ($p>0,05$). 5. ve 6. sınıf öğrencilerinin puan ortalamaları arasında fark olup olmadığı da Mann-Whitney U testi ile sınanmıştır. Benzer şekilde, Scratch ile programlama öğretiminde öğrencilerin problem çözme becerileri sınıf seviyesine göre anlamlı bir farklılık göstermemiştir ($p>0,05$).

Scratch ile programlama öğretiminde öğrencilerin yapılandırılmamış problemleri çözerken uyguladıkları işlem adımları nelerdir?

Öğrenciler Scratch ile oyun tasarımında ağırlıklı olarak sırasıyla; oyunu zihinlerinde gerçek hayat şartlarına uygun olarak kurgulamaktadırlar, kurguladıkları oyunun tasarımını metne dökmektedirler, tasarıma uygun olarak kod bloklarını yerleştirmektedirler, programı çalıştırıp bir sonraki aşamaya geçmektedirler ve hata durumunda ilgili bölümdeki kod bloklarını tasarıma göre kontrol etmektedirler.

TARTIŞMA VE SONUÇ

Bu çalışma, öğrencilerin problem çözme becerilerinin problem çözme adımları üzerinden araştırılması amacıyla gerçekleştirilmiştir. Scratch ile yapılan programlama öğretiminin sonucunda, öğrencilerin problem çözme becerilerinin daha önceki çalışmalarda da bulunduğu gibi olumlu yönde etkilendiği (Akcaoglu ve Koehler, 2014; Brown vd., 2008; Calder, 2010; Nam, Kim ve Lee, 2010; Shin ve Park, 2014) sonucuna ulaşılmıştır. Ayrıca, problem çözme becerileri sınıf seviyesine ve cinsiyete göre farklılık göstermemektedir. Kalelioğlu ve Gülbahar'ın (2014) araştırma sonuçlarına göre, 5 haftalık birer saatlik uygulama sonucunda, aynı sınıf seviyesindeki öğrencilerin öntest ve sontest problem çözme beceri puan ortalamaları arasında anlamlı bir farklılık olmadığı bulunmuştur. Bu sonucun eğitim süresinin kısalığından ve örneklemin küçüklüğünden kaynaklanabileceğini ifade etmişlerdir. Karabak ve Güneş'in (2013) çalışmasında belirtildiği üzere Bilişim Teknolojileri ve Yazılım dersinin öğretim programında Problem Çözme, Programlama ve Özgün Ürün Geliştirme alanının yıllık plandaki süresi en az 11 haftalık 22 ders saati süresine denk gelmelidir. Bizim araştırmamızda anlamlı farklılık çıkmasının sebebi eğitimin 22 ders saatini dolduracak şekilde verilmesi ve örneklem büyüklüğünün yeterli olmasıdır.

Robertson (2012)'a göre; oyun oynamaya kıyasla, oyun tasarım süreci karmaşık tasarım görevleri içeren bilişsel bir süreçtir. Tasarım görevleri, tam ve işlevsel bir sistem oluşturmak için pek çok birbiri ile ilişkili değişkenin ve parametrenin kullanılmasını gerektirir. Öğrenciler, Scratch programında oyun tasarım sürecinde koşullu ifadeleri ve döngüleri (kompütasyonel ifadeleri) kullanmak zorunda kalmışlardır. Dolayısıyla ilgili alanyazında (Adams, 2010; Adams ve Webster, 2012; Claypool, 2013) gösterildiği gibi, bu çalışmada da oyun tasarımı projelerinin, öğrencileri programlama deyimlerini öğrenmeye zorladığı bulunmuştur.

Yorumlama ve problemi tanımlama aşamasına baktığımızda öğrencilerin oyunu zihinlerinde gerçek hayatı referans alarak kurguladıkları ve bu kurguladıkları oyunun tasarımını metne döktükleri görülmektedir. Öğrenciler, oyunu zihinlerinde tasarlamadan direkt uygulamaya yönelik kod bloklarını birleştirmeye geçtiklerinde zorlanacaklarını ifade etmişlerdir. Gomes ve Mendes (2007) programlama öğrenmenin güçlükleri ve çözümlerini araştırdıkları çalışmalarında öğrencilerin birçok kez, problemi tam anlamıyla anlamadan problemin çözümüne yönelik eyleme geçtiklerini vurgulamıştır. Öğrenciler, Scratch ortamında yapılandırılmamış oyun tasarım görevlerini ilk önce zihinlerinde kurgulayarak ve tasarım sürecini metne dökerek problemi tanımlamaktadırlar. Bu çalışmada, alanyazında (Calder, 2010; Lee, 2011; Resnick vd., 2009) belirtildiği gibi öğrencilerin x ve y ekseni gibi matematiksel ifadeleri de öğrendikleri görülmüştür. Ayrıca, öğrencilerin, oyunu zihinlerinde gerçek hayat şartlarına uygun olarak kurgulamalarının hayal gücünün gelişimini desteklediği bulunmuştur ve bu alanyazındaki bulgularla örtüşmektedir (Burke ve Kafai, 2010; Giannakos, Jaccheri ve Proto, 2013; Wilson ve Moffat, 2010).

Öğrenciler, çözümler geliştirme ve çözüm sürecini izleme aşamasında kod bloklarını oyunun tasarımına göre yerleştirmektedirler. Öğrencilerin referans olarak aldıkları bir tasarım vardır. Her öğrencinin problemin çözümüne yönelik oyun tasarımı farklılık göstermektedir. Öğrenci görüşlerinden anlaşılacağı üzere; yapılandırılmamış problemlerin, belirli bir karar alma süreci kullanılarak saptanabilen tek bir çözüm yolu yoktur ve bu bulgu alanyazınla da uyuşmaktadır (Chi ve Glaser, 1983; Jonassen, 1997). Scratch ile oyun programlama sürecinde her öğrenci kendi temposunda öğrenmekte (Gomes ve Mendes, 2007) ve aşamalı olarak kendi bilgisini oluşturmaktadır. Öğrenciler, aynı zamanda, bu aşamada kontrol menüsündeki kompütasyonel ifadelere karşılık gelen kod bloklarının önemine dikkat çekmiştir. İlgili alanyazına (Adams ve Webster, 2012; Brown vd., 2008; Calder, 2010; Claypool, 2013) baktığımızda Scratch programının kompütasyonel kavramları öğretmekte başarılı olduğunu görmekteyiz. Oyun tasarım görevleri kompütasyonel kavramların öğrenilmesinde içsel motivasyonu arttırmaktadır.

Karar verme ve değerlendirme aşamasında öğrenci görüşlerini incelediğimizde; öğrencilerin çoğunluğu oyunu doğru tasarlamaktadır, hatalar çoğunlukla kod bloklarını tasarıma uygun birleştirememelerinden kaynaklanmaktadır. Bunun sebebi olarak da tasarımı detaylandırırken kompütasyonel ifadelere uygun olarak tasarlamadıkları gösterilebilir. Dolayısıyla, mantıksal ve koşullu ifadelerin yani kompütasyonel ifadelerin karşılığını Scratch programındaki Kontrol menüsünde bulmakta zorlanmaktadırlar. Yapılandırılmamış problemlerin çözüm süreci tasarım süreci olarak düşünülebilir. Öğrencilerin kontrol menüsündeki kompütasyonel ifadelere uygun olarak oyunu tasarlamaları gerekmektedir. Bu süreç ise bilgi işlemsel süreç (Aho, 2012; Barr, Harrison ve Conery, 2011; Michaelson, 2015; Wing, 2006) olarak tanımlanabilir.

Kavramsal olarak yapılandırılmamış problemleri çözmek, problem çözümü için sistematik olarak bir araştırma değil de bir tasarım süreci olarak düşünülebilir. Öğrencilerin oyunu zihinlerinde gerçek hayat şartlarına uygun olarak kurgulamaları, kompütasyonel ifadelere uygun olarak tasarlamaları ve detaylandırmaları problemin çözümünü kolaylaştırmaktadır. Öğrenciler Scratch ile oyun tasarımında ağırlıklı olarak sırasıyla; oyunu

zihinlerinde gerçek hayat şartlarına uygun olarak kurgulamaktadırlar, kurguladıkları oyunun tasarımını metne dökmektedirler, tasarıma uygun olarak kod bloklarını yerleştirmektedirler, programı çalıştırıp bir sonraki aşamaya geçmektedirler ve hata durumunda ilgili bölümdeki kod bloklarını tasarıma göre kontrol etmektedirler. Öğrencilerin hata durumunda, ilgili bölümü tasarıma göre kontrol etmeleri; oyunu kompütasyonel ifadelere uygun tasarlamalarının önemini göstermektedir (Aho, 2012; Barr vd., 2011; Michaelson, 2015; Wing, 2006).

Öğrenci görüşlerinden de anlaşılacağı üzere her öğrencinin bilgi işlemsel düşünme süreci farklılık göstermektedir. Bu durum, günümüzde bireylerin farklı düşünme biçimlerinin farkında olmasının ve problem çözümüne giden yolda bilgisayar mantığıyla kararlar almasının; yani, bilgi işlemsel düşünmesinin (Barr vd., 2011; Wing, 2006) gerekli olduğunu göstermektedir. Katılımcıların programlama öğretiminde kullanılan Scratch yazılımının çok yönlü düşünmeye ve alternatif çözüm yollarının oluşturulmasına katkıda bulunabileceğine ilişkin görüşleri, Scratch yazılımının bilgi işlemsel düşünmeyi bu açılardan destekleyebileceğini göstermesi açısından önemlidir.

ÖNERİLER

Ortaya çıkan sonuçlara göre; problem çözme becerilerine sahip öğrenciler yetiştirmek için alternatif bir metot olarak programlama öğretirken Scratch ile oyun tasarlanması önerilmektedir. Öğrencilerin problem çözme adımlarını incelediğimizde, Scratch ortamında yapılandırılmamış problemleri çözebilmeleri ve oyun projesini başarı ile tamamlamaları için tasarım odaklı yaklaşım uygulanması önerilmektedir. Yorumlama ve problemin tanımlanması aşamasında öğrencilere rehberlik sağlanması açısından, sınıf ortamında oyundaki kuklaların öğrenciler ile gerçek hayatı referans alarak canlandırılması öğrencilerin zihinsel süreçlerini hızlandıracak ve kolaylaştıracaktır. Öğrenciler, Scratch Kontrol menüsündeki kompütasyonel ifadelere uygun olarak oyunun tasarımını detaylandırırken öğretmenlerin rehberlik etmeleri önerilmektedir.

Programlama konusu ile ilgili yayınlarda 2012 yılından bu yana istikrarlı bir artış görülse de bu yayınlar kodlama öğretiminin pedagojisine yoğunlaşmamaktadır. Scratch ile oyun programlama sürecindeki öğrencilerin kullandıkları işlem adımları kodlama öğretiminin pedagojisine yönelik öğretmenlere rehberlik edebilir. Scratch programı iki boyutlu görsel bir programlama aracı olduğundan, çalışmanın derinlik hissi veren ve böylece daha gerçekçi olabilecek üç boyutlu görsel bir programlama aracı olan Alice programı kullanılarak uygulanması önerilebilir. Ayrıca, gerçek deneysel desenler ile araştırmalar gerçekleştirilmesi önerilmektedir.

Anahtar Kelimeler: Scratch, problem çözme becerisi, programlama öğretimi, kodlama, ortaokul öğrencileri.

REFERENCES

Adams, J. C. (2010). "Scratching Middle Schoolers' Creative Itch", Proceedings of the 41st ACM Technical Symposium on Computer Science Education, March 10–13, Milwaukee.

Adams, J. C., & Webster, A. R. (2012). "What Do Students Learn About Programming From Game, Music Video, And Storytelling Projects?", Proceedings of the 43rd ACM Technical Symposium on Computer Science Education, 29 March, Raleigh.

Aho, A. V. (2012). "Computation and Computational Thinking". *The Computer Journal,* 55(7): 832-835.

Akcaoglu, M., & Koehler, M. J. (2014). "Cognitive outcomes from the Game-Design and Learning (GDL) after-school program". *Computers & Education,* 75: 72-81.

Barr, D., Harrison, J., & Conery, L. (2011). "Computational Thinking: A Digital Age Skill for Everyone". *Learning & Leading with Technology,* 38(6): 20-23.

Brown, Q., Mongan, W., Kusic, D., Garbarine, E., Fromm, E., & Fontecchio, A. (2008). "Computer Aided Instruction as a Vehicle for Problem Solving: Scratch Programming Environment in the Middle Years Classroom", ASEE Annual Conference & Exposition, June 22-25, Pittsburgh.

Burke, Q., & Kafai, Y. B. (2010). "Programming & Storytelling: Opportunities for Learning About Coding & Composition", Proceedings of the 9th International Conference on Interaction Design and Children, June 9–12, Barcelona.

Büyüköztürk, Ş., Çakmak, E. K., Akgün, Ö. E., Karadeniz, Ş., & Demirel, F. (2017). *Bilimsel Araştırma Yöntemleri*. Ankara: Pegem Akademi.

Calder, N. (2010). "Using Scratch: An Integrated Problem-solving Approach to Mathematical Thinking". *Australian Primary Mathematics Classroom,* 15(4): 9-14.

University of Pittsburgh (1983). *Problem Solving Abilities*. Pittsburgh: L. R. A. D. Center.

Claypool, M. (2013). "Dragonfly: strengthening programming skills by building a game engine from scratch". *Computer Science Education,* 23(2): 112-137.

Coşkun, M. (2004). *Coğrafya Öğretiminde Proje Tabanlı Öğrenme Yaklaşımı.* Yayınlanmamış Doktora Tezi, Gazi Üniversitesi Eğitim Bilimleri Enstitüsü, Ankara.

Creswell, J. W., & Clark, V. L. P. (2007). *Designing and Conducting Mixed Methods Research*. Thousand Oaks, California: Sage Publications.

Ge, X. (2001). *Scaffolding Students' Problem-solving Processes on an Ill-Structured Task Using Question Prompts and Peer Interactions.* Doctoral dissertation, The Pennsylvania State University, Pennsylvania.

Giannakos, M. N., Jaccheri, L., & Proto, R. (2013). "Teaching Computer Science to Young Children through Creativity: Lessons Learned from the Case of Norway", Proceedings of the 3rd Computer Science Education Research Conference on Computer Science Education Research, 4-5 April, Arnhem.

Gomes, A., & Mendes, A. J. (2007). "Learning to Program-difficulties and solutions", International Conference on Engineering Education, September 3–7, Coimbra.

Jonassen, D. H. (1997). "Instructional Design Models for Well-structured and III-structured Problem-solving Learning Outcomes". *Educational technology research and development,* 45(1): 65-94.

Jonassen, D. H., & Kwon II, H. (2001). "Communication Patterns in Computer Mediated Versus Face-to-Face Group Problem Solving". *Educational technology research and development,* 49(1): 35-51.

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.

Karabak, D., & Güneş, A. (2013). "Ortaokul Birinci Sınıf Öğrencileri Için Yazılım Geliştirme Alanında Müfredat Önerisi". *Eğitim ve Öğretim Araştırmaları Dergisi,* 2(3): 163-169.

Lee, Y.-J. (2011). "Scratch: Multimedia Programming Environment for Young Gifted Learners". *Gifted Child Today,* 34(2): 26-31.

Malan, D. J., & Leitner, H. H. (2007). "Scratch for Budding Computer Scientists", Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education, March 7–10, Covington.

Michaelson, G. (2015). "Teaching Programming with Computational and Informational Thinking". *Journal of Pedagogic Development,* 5(1): 51-66.

Milli Eğitim Bakanlığı [MEB] (2012). *Bilişim Teknolojileri Ve Yazılım Dersi (5, 6, 7 Ve 8. Sınıflar) Öğretim Programı*. Retrieved from http://ttkb.meb.gov.tr/program2.aspx/program2.aspx?islem=1&kno=196

Nam, D., Kim, Y., & Lee, T. (2010). "The Effects of Scaffolding-Based Courseware for The Scratch Programming Learning on Student Problem Solving Skill", Proceedings of the 18th International Conference on Computers in Education, November 29 - December 3, Putrajaya.

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

Robertson, J. (2012). "Making games in the classroom: Benefits and gender concerns". *Computers & Education,* 59(2): 385-398.

Scratch. (2018, January 14). *Scratch – About.* Retrieved from https://scratch.mit.edu/about

Shin, S., & Park, P. (2014). "A Study on the Effect affecting Problem Solving Ability of Primary Students through the Scratch Programming". *Advanced Science and Technology Letters,* 59: 117-120.

Trilling, B., & Fadel, C. (2009). *21st century skills: Learning for life in our times*. San Francisco: John Wiley & Sons.

Voss, J. F., & Post, T. A. (1988). The Nature of Expertise. *On the solving of Ill-structured problems* (261-285). New York: Psychology Press.

Wilson, A., & Moffat, D. C. (2010). "Evaluating Scratch to introduce younger school children to programming", Proceedings of the 22nd Annual Psychology of Programming Interest Group, September 19-21, Madrid.

Wing, J. M. (2006). "Computational Thinking". *Communications of the ACM,* 49(3): 33-35.

Yıldırım, A., & Şimşek, H. (2013). *Sosyal Bilimlerde Nitel Araştırma Yöntemleri*. Ankara: Seçkin Yayıncılık.