IJQE

International Journal of Quality in Education

Online, https://dergipark.org.tr/tr/pub/ijqe

Volume: 5, Issue 2, 2021

e-ISSN:2636-8412

SCRATCH IN TEACHING PROGRAMMING: EFFECT ON PROBLEM SOLVING SKILL AND ATTITUDE

Rukiye Berna Bala (Ministry of National Education, Turkey)
r.berna.bala@gmail.com
0000-0003-4458-8320

Prof. Dr. Füsun Gülderen Alacapınar (Necmettin Erbakan University, Turkey)
fusunmireli@yahoo.com
0000-0001-7515-385

Abstract

The study seeks to identify the effect of the Scratch program used in the teaching programming language to 6th-grade students on student performance, problem-solving skills, and their attitude towards the course on information technologies and to compile student opinions on its implementation. 22 students took part in the study. The study used pre and post-test single-group experimental design. It took 8 weeks. The first five weeks were allocated activities related to behavior in information and concept building. The last three weeks were for behavior related to practice. Students were asked to develop a project that covers objectives mentioned and of interest to them. Achievement test, problem-solving scale, scale on student attitude towards information technologies course were data collection tools used and an interview form was used to take the opinions of students. At the end of the study, it was observed that the post-test achievement scores of students were higher. It was found that the Scratch program had no statistically significant effect on students' problem-solving skills while there is an increase in post-test scores related to attitude towards the course. It can be said that student opinions on the use of Scratch in a teaching programming language are in general positive.

Keywords: *Attitude; Programming; Problem-solving skill; Scratch*

Introduction

Each citizen needs education in computer science to grasp and effectively use the relevant technology and to produce what is intended. This education should not intend to accord individuals such professional titles as "computer scientist" or "software expert". The idea is to provide opportunities to people, including younger age groups in the first place, to use a computer to develop new ideas and to produce. To this end, there are now many countries adding computer science as a course to their curricula (Sayın, 2017).

In recent years there is mention of programming skills as a distinguishing and desired competence in many areas (Kalelioğlu & Gülbahar, 2014; Yükseltürk & Altıok, 2015). There are now various types of software developed and activities carried out to introduce programming particularly to younger age groups and build relevant skills. Programming is a difficult and long process of education.

The teaching program in the course Information Technologies was updated in 2012 and the name of the course was changed to 'Information Technologies and Software". Also, a unit on programming was added (MoNE, 2012). The teaching program was updated once more in 2018 with the addition of the unit "Problem Solving and Programming". This unit envisages information and skill-building in the following: algorithm design (searching, sequencing, vb.); verbal and visual expression, assigning variables for problem-solving, sequential logic, decision structure, use of circuit and function structure, selection of appropriate programming approach in problem solving and skills in practice (MoNE, 2018).

Examining related studies we find that when they take courses in programming students may have higher motivation for their schools and courses and their problem solving and analytic thinking skills may improve (Akpınar & Altun, 2014). In a study investigating the effect of using a robot in coding education on secondary school students' problem-solving skills Özer (2019) statistically significant difference was found in scores of experimental group students in problem-solving inventory. Another study states that programming skills, independent of the programming language used, may

build skills in problem-solving and analytical thinking in different fields such as logical thinking and algorithm setting (Ersoy, Madran & Gülbahar, 2011). Çam (2019) says robot-supported programming education creates a significant difference in motivation levels of students in the experimental group.

One issue to consider is the ages at which this education is to be given. Utting, Cooper, Kölling, Maloney, and Resnick (2010), for example, present different age groups for different software and suggest the age interval 8-16 for Scratch. They also add that the level of cognitive development of children will be influential in selecting software to be used in programming education.

There are many factors affecting achievement in programming language courses. Examples include motivation, attitude towards programming, complex nature of programming language, and teaching method and design. In programming education, if the student is not motivated enough he or she may get frustrated as the course is in progress and this may lead to lesser and lesser studying. Students who think there has to be a lot of computer work in developing a program may find this boring (Ersoy et. al., 2011).

One situation troubling new beginners in programming is misspelling made while writing codes. In this process which closely resembles learning a new language misspelled words and missed signs trouble students quite a bit. It is essential to grasp basic concepts in programming education. Otherwise, programs developed to turn into performances in rote fashion. Here, the first step must be teaching the logic of the algorithm as the basis, and then the learning basic concepts must follow. The course must proceed with tools to facilitate the process and make it more fun.

Taking a look at relevant studies we see that various types of software are used for beginners to make the education process more fun and to ease the learning of basic concepts. (Coşar, 2013; Genç & Karakuş, 2011; Yükseltürk & Altıok, 2016). Scratch, Microsoft Small Basic, Alice, and MIT AppInventor can be mentioned as programs making the process more interactive and easier. In a study examining the opinions of candidate teachers in information technologies who received programming education through Scratch, Yükseltürk and Altıok (2016) found that the perception of candidate teachers related to the Scratch program was positive.

Education in Programming

Taking a look at studies on programming education around the world it can be observed that countries recently attach importance to this education. It can be said that this education is often included in secondary education curricula with recent insertion into primary school curricula as well. According to the European Schoolnet study report, there are European Union countries adding programming as a course to their school curricula. Further, some countries introduce programming to their students through extracurricular activities (Şimşek, 2018).

Recently various activities are organized throughout the world to introduce programming education to children at younger ages. Studies conducted so far suggest that there is consensus on the possibility of building such 21st-century skills as problem-solving, creativity, algorithmic thinking, and information processing in individuals through the teaching of programming and computer sciences (Çatlak, Tekdal & Baz, 2015).

The site code.org site as a voluntary initiative is active with the motto "Anybody can learn". Activities on the site under the title "Coding Hour" aim to introduce coding to students. There is language support in activities that can be accessed online and activities make use of various films, animations, and characters that children and youth find interesting.

The EU CodeWeek organization is carried out by the European Schoolnet whose members include the Ministry of National Education (MoNE). The activity taking place in October covers schools in Europe who want to participate. The EU CodeWeek explains the objective of its activities as a grassroots movement that encourages creativity, problem-solving, and cooperation through programming and other technological activities. The objective here is to make programming more visible and demonstrate to youth, adults, and the elderly how ideas can be translated into life by coding. The organization also seeks to uncover these skills and bring motivated persons together for learning (https://codeweek.eu/about/).

We observe different countries adding courses in computer and coding to curricula of their primary and secondary education schools. England, for example, changed the name of the course which used to be Information and Communication Technologies to "Computing" in 2013. This computing course is built upon three pillars as computer

science, information technologies, and digital literacy. England is the first G20 country that made computer science courses compulsory in primary education (Sayın, 2017).

In Turkey, programming education is delivered at the secondary education level, under information technologies in vocational high schools and various engineering departments of universities. There are also special courses for those who want to advance in this area. Mostly in laboratories, courses are delivered through the demonstration technique. The assessment is done by written exams, practice tests, or graduation projects. At the primary education level, the objective of programming education is not professional software development but to help students learn by applying their algorithmic thinking skills in concrete ways. There are many instruments like Scratch, Alice, and Kodu that are designed to change students' attitudes to programming in a positive way (Avcı & Ersoy, 2018: 74).

Programming Tools for Children

Given the specific words and rules and abstract concepts of programming languages children may face difficulties at early stages. There are many block-based programming tools developed to make the process easier for beginners.

Instead of the syntax used in block-based programs, there are blocks expressed by visuals that work with a drag-and-drop basis. These are for avoiding misspellings and memorizing rules that are valid for text-based languages. The puzzle-like design of blocks allows only for bringing correct pieces together. Studies show block-based instruments contribute to students' learning of programming. In a study examining the effect of programming education with a scratch on the motivation and achievement of students, Erol (2015) finds a significant difference in programming achievement scores in favor of participants in the experimental group. Students working with Scratch say activities are easy and fun and functional in building programming logic and enhancing motivation. There are various programs developed for children in programming education.

Scratch Programme

The Scratch program announced with the slogan "Imagine Code Share" was developed in the media laboratories of MIT (Massachusetts Institute of Technology). The program is now used with more than 40 languages in some 150 countries. With Scratch,

students develop programs by dragging and dropping code blocks instead of writing codes. With this program that can be used not only by children but all who have just started coding you can develop your stories, plays, or animations and share them with others on the site. Scratch is a visual programming environment (firstly for the age group 8-16) that helps its users learn programming while working on their projects like animated stories and games (Maloney et. al., 2010). It is a black-based programming tool that is free and usable by people of all ages.

Developing projects by bringing code blocks together with the Scratch program is much easier than in text-based programming languages (Genç & Karakuş, 2011). Minimizing possible spelling mistakes in writing the program, the Scratch program is an appropriate program for children in this respect. Alp (2019) finds that the Scratch program improves secondary education students' attitude to computer and problem-solving skills. This program also offers opportunities to children for developing projects in different fields. The Scratch facilitates teachers' work to teach programming concepts at all levels and to develop products that also provide them opportunities to motivate their students (Yükseltürk & Altıok, 2016).

It is stated that while programming and sharing interactive projects Scratch users can think about and learn ways of creative, systematic, and collaborative thinking which are considered essential skills of the 21st century (Resnick et. al., 2009). The reason for using the Scratch program in this study is that it has its Turkish language support, availability of many source documents, beings without any fee, and that the method of drag-and-drop is more suitable to the level of new beginners in programming.

Problem Solving Skill and Information Technologies and Software Course

The following are some points stressed under the heading "specific objectives" in the updated teaching program of the course Information Technologies and Software (MoNE, 2018):

- Ensure students' building of an overall understanding and technical background in computer science,
- Ensure students' acquisition and development of problem-solving and information processing skills,
- Ensure that students can follow and evaluate reasoning processes.

With these, students are expected to seek and question information, use information technologies by complying with relevant ethic rules, solve problems they face in the process, and enter into effective communication. It is targeted, with courses in Information Technologies or Computer Sciences to be delivered in primary and secondary schools and high schools to train students as individuals capable of keeping track of innovations, understanding, and questioning the world they live in. In this context, different institutions set various standards to determine competencies that students are expected to have. The International Society for Technology in Education (ISTE) set some standards for students, teachers, teachers, and managers information technologies on the use of education technologies. The standards set for students are as follows (ISTE, 2016): Empowered Learner, Digital Citizen, Knowledge Constructor, Innovative Designer, Computational Thinker, Creative Communicator, and Global Collaborator. Under the heading "Computational Thinker" students are expected to develop and employ strategies for understanding and solving problems in ways that leverage the power of technological methods to develop and test solutions.

In Turkey, students are offered Computer Science courses in high schools and Information Technologies and Software courses in secondary schools. It is desired that students can use information and communication technologies effectively, efficiently, and in line with ethical rules. As stated in the renewed teaching program of the course the objective is to train students who are capable of using technology effectively, suggesting solutions to problems, and developing their software to produce information. In secondary schools, the course comprises five units (MoNE, 2018): Information Technologies, Ethics and Safety, Communication Research and Cooperation, Product development, Problem Solving, and Programming.

The present study investigates the use of Scratch software in secondary education to make programming education more effective at that level and to teach programming. Another objective is to build awareness in students related to this field. In younger age groups students may be oriented to this field by finding out about their interest or skills in programming. Paths can be opened for students who want to advance further in the field. It can be said the programmers well trained in information and communication technologies will contribute to the national economy.

Problem Sentence

Does the scratch program used in the teaching programming language to secondary school 6th-grade students have any significant effect on student scores in achievement, problem-solving skills, and their attitude towards the course? What are the opinions of students on teaching programming through scratch?

Hypotheses

1) There is a significant difference between pre and post-test scores of the group to which the scratch program was applied.
2) There is a significant difference between pre and post-test scores of the group to which the scratch program was applied concerning the problem-solving skills scale.
3) There is a significant difference between pre and post-test scores of the group to which the scratch program was applied concerning attitude towards the course.
4)

Sub-problem

What are the opinions of students about teaching programming with Scratch?

Research Design

The study investigates the effect of the Scratch program used in courses on students' scores in achievement, problem-solving skills, and attitude towards the Information Technologies course by using the methods of pre and post-test, experimental group design, and qualitative survey. The "descriptive approach" was adopted. Students' opinions on programming education through Scratch were also solicited. Tests were applied to the student after the completion of experimental work.

Study Group

An appropriate working group was used in the study instead of determining any universe and sample. The working group of the study consists of 6th-grade students who were taking an Information Technologies course in a state school in Seydişehir District of Konya Province. 22 students took part in the study.

Measurement Tools

Achievement Test: The test used in determining the achievement level of students was prepared by the researcher. The achievement test contained questions of different types which were formulated upon the review of textbooks, relevant sites, and various academic studies. 30 such questions were formulated. Opinions of three experts were

taken to check the relevance of these questions. The reliability of the study was found as 79% by using Miles-Huberman's (1994) coder reliability percentage. The reliability coefficient in the study was calculated as 0.814.

Problem Solving Scale: The Problem Solving Scale used in the study was borrowed from the following source: Kardaş et. al. (2014), "Adapting Problem Solving Inventory to Primary Education Students: Outcomes of Confirmatory Factor Analysis." The study mentioned the adaptation to 5^{th}-grade students of PSS as a psychometric tool for measuring the problem-solving skills of adults as translated into Turkish by Şahin, Sahin, and Heppner (1993). The Cronbach Alfa reliability of the scale was calculated and the coefficient was found as 0.608.

Scale for Attitude towards Information Technologies Course: The "Scale for Attitude towards Information Technologies Course" was used at the beginning and end of the study to see whether there is any significant change in student attitude to the course. This scale used in the study is the attitude scale developed by Işık (2010) in a doctoral dissertation. The Cronbach Alfa reliability of the scale was calculated and the coefficient was found as 0.945.

In the qualitative part of the study opinions of students in the working group on implementation were taken. Descriptive analysis was applied to these opinions followed by explanations and comments. Descriptive analysis is the simplest form of analysis used in analyzing qualitative data. In this method, the researcher must gather questions under some headings and categories in reporting instead of just writing them down as they are (Sönmez & Alacapınar, 2014). The semi-structured questionnaire consists of open-ended questions formulated by the researcher.

The study lasted for eight weeks in total with two class sessions in a week each taking 40 minutes. Class session plans were developed and implemented in line with targeted behaviors set for the study. During the first five weeks work concentrated on behavior related to information and comprehension levels. Students were informed about basic concepts in programming, linear logic, cycle logic, and decision. Relevant case studies were demonstrated to students. Students were then asked to launch their model practices. Students were guided during their practices. Activities carried out were as follows:

- Direction Keys

- Aquarium
- Hunter Shark
- Ball Game

The last three weeks were devoted to behavior at the practice level. Students were asked to develop a project that is of interest to them and covers the targets mentioned. Semi-structured interview forms were used at the end to solicit students' opinions. Then post-tests were applied to the working group in achievement, problem-solving scale, and attitude scale.

Data Analyses

In the achievement test applied, 1 and 0 were scores assigned to correct and incorrect answers, respectively. In the test the 1. question has eight and the 2. question has seven sub-items. The maximum score that can be obtained is 43. In order to determine whether there is a significant difference between pre and post-test scores of students in achievement, problem-solving scale and attitude towards Information Technologies course paired samples t-test was applied. To assess the opinions of students, descriptive analysis was applied to data obtained by using a semi-structured interview form.

Findings

1. Findings Related to the First Hypothesis: Findings related to whether there is a significant difference between students' pre and post-test scores in achievement are examined and shown in Table 1.

Table 1. Achievement Test Pre And Post-Test Related Samples t-Test Results

	N	$\bar{X}$	SS	*T*	P
Pre-test	22	11.14	4.357	-9,893	0,00*
Post-test	22	23.82	7.248		

*$p<0.05$

Paired samples t-test was applied to see if there is a significant difference between pre and post-test scores of students in achievement. The difference between pre-test scores ($\bar{X}_{öntest}$ = 11.14) and post-test scores ($\bar{X}_{sontest}$= 23.82) was calculated as -9.893 at 21 degrees of freedom [$t_{(22)}$ = -9.893, p= 0.00 < 0.05]. Looking at the P-value we see a significant difference at a level of 0.05. This indicates there is a significant difference between pre and post-test scores of students.

2. Findings Related to the Second Hypothesis: Outcomes of the sample t-test applied to see whether there is a significant difference between students' pre and post-test scores in problem-solving skills test are given in Table 2.

Table 2. Problem Solving Scale Pre and Post-Test Related Samples t-Test Results

	N	$\bar{X}$	SS	*T*	P
Pre-test	22	54.5	6.54	-.840	0.410*
Post-test	22	55.9	6.60		

*p<0.05

Examining the averages of pre and post-test scores of the working group in the problem-solving scale in Table 2 we see that the average of post-test scores ($\bar{X}$= 55.9, SS=6.60) is higher than the average of pre-test scores ($\bar{X}$= 54.5, SS=6.54). According to paired samples t-test, the *t* value in scores obtained from problem solving scale was calculated as - .840 at 21 degrees of freedom ($t_{(22)}$ = -,840, p= 0,410 <0,05). Looking at the P-value we don't see any significant difference at the level of 0.05 which suggests that the scratch program used in programming education has no statistically significant effect on the problem-solving skills of students.

3. Findings Related to the Third Hypothesis: To determine the effect of the Scratch program used in programming education on the attitude of students towards the Information Technologies course, an attitude scale was distributed before and after implementation. The outcomes of paired samples t-test applied to see whether there is a significant difference between average scores are given in Table 3.

Table 3. Attitude Scale to Information Technologies Course Pre and Post-Test Related Samples t-Test Results

	N	$\bar{X}$	SS	*T*	P
Pre-test	22	70.0	3.28	-5.002	0.000*
Post-test	22	89.5	18.5		

*p<0.05

Examining averages of pre and post-test scores of the working group in attitude towards Information Technologies course scale in Table 3 we see that the average of post-test scores ($\bar{X}$= 89.5, SS=18.5) is higher than the average of pre-test scores ($\bar{X}$= 70, SS=3.28). Also, according to paired samples t-test, the *t* value in scores obtained from the attitude towards the course scale was calculated as 5.002 ($t_{(22)}$ = -5.002, p= 0,00 <0.05) at 21 degrees of freedom. Looking at the p-value we see a significant difference at the level of 0.05 which suggests that the Scratch program used in programming education contributes positively to student attitude towards Information Technologies course.

4. Findings Related to the Sub-problem

It can be said that students have a positive opinion about the Scratch program. They say they had fun and a nice time while using the program. Some comments made by students are "We can create fine games", "I liked it, it must be in all classes", "It contributed to my problem solving, I participated more in my class."

4.1 Opinions of Students on Cases They Found Interesting in their Work with Scratch Programme in Problem Solving and Programming Unit

Students say they found their work with the Scratch program interesting. One student says, "I can change the scene and characters as I wish, I can do anything I want…" Comments by some other students include: "What catches my interest is that we can do animation and cartoon films", "It does what I tell it to do"… Two students were different from others and they found nothing interesting for them in the program.

4.2 Opinions of Students on Cases They Disliked in their Work with Scratch Programme in Problem Solving and Programming Unit

Students say they had no dislike or trouble point in their work with the Scratch program. One student says "There was nothing posing difficulty" and another "There was nothing that I disliked". Still, some students say it is annoying to have difficulty in finding the mistake they have done. One student says, "There were times I got confused" and another: "It was a problem when what I did turned out to be incorrect and I could not find where the mistake was."

4.3 Opinions of Students on Benefits of Problem Solving and Programming Unit

In general, students say they liked the course more, learned new things, and could use the computer more effectively. One student said, "I realized I could use the computer in a more useful way." Another said the following about the work done: "I learned how to make games to help in problem-solving." A student: "It was useful for the course, now we can use computers more effectively." Other comments by students are as follows: "We can earn some money by developing our software"; "It added to our knowledge and helped us in having fun at the same time"; "I learned how to use the computer"; "We've learned programming; "It made the course more enjoyable."

Discussion

Outcomes of the study show that work conducted had positive effects on students' academic achievement. These outcomes are in line with the findings of the study by Özoran, Çağıltay, and Topallı (2012) in the same field. They found that after the use of Scratch programming environment in the course "Introduction to Programming" the number of failing students reduced compared to the previous year. Korkmaz (2016) investigated the effects of Scratch-based game activities on students' attitudes towards learning computer programming, perception of self-efficacy, and academic achievement. It is stated that the achievement of the group that taught C++ courses with Scratch is higher than the group taught by using conventional methods. Dinçer (2018) examined the effect of programming education with Scratch and Kodu Game Lab on the academic performance of secondary education 6th-grade students, their attitude, and their perception of self-efficacy. Dinçer concluded that the achievement of students learning to program with Scratch is significantly higher than others learning it by Kodu Game Lab. Studies conducted so far suggest that applied teaching significantly affects student achievement since they learn the program by doing it in practice. Also, examining the qualitative findings of the study we see that students mentioned their enjoyment of the course. This enjoyment is one of the factors affecting students' performance in a course. Students say they can earn money with what they have learned. Students may be much more willing to learn when they feel what they are learning will have some use in their lives. This can be taken as another factor affecting performance.

Another point examined in the study was the effect on problem-solving skills. Here, no significant difference could be found between pre and post-test scores of students. This outcome is in line with the findings of Genç and Tınmaz (2010) and Kalelioğlu and Gülbahar ın (2014) in their studies. Korkmaz (2018) too found game activities performed with Scratch had no significant contribution to problem-solving skills. However, there are also some other studies reaching just the opposite conclusions. Nam, Kim, and Lee (2010) arranged a process to facilitate teaching in programming with Scratch that was in line with the scaffold approach and examined its effect on students' problem-solving skills. They found significant improvement in problem-solving skills in both groups. Çetin (2012) concludes that programming education enhances problem-solving skills in children. However, students could not use skills they have gained in practice in some new situations. This may be due to limited time and the weight accorded to achievements in practice during class sessions. Another reason can be not solving some practical problems fully.

The study examined the effect of Scratch on student attitude towards the course. Here there was a significant difference between pre and post-test scores of students on the attitude scale. Coşar (2013) found positive effects on student attitude towards the computer. Ouahbi et. al. (2015) investigated the effect of the Scratch program they use in teaching the basics of programming on student motivation for programming. They found there is a significant increase in the achievement and motivation of students in the group using the Scratch program relative to students in the other group. Qualitative findings of the study also support this as seen in relevant comments made by students. Students supported the affective domain by stating their enjoyment of the course, the usefulness of what they learned, and possibilities of earning money with these skills.

In conclusion, there is improvement in the academic achievement of students receiving programming education with scratch while there is no significant effect on problem-solving skills. Also, there is a positive change in their attitude towards the course Information Technologies.

In the light of the findings of the study, the following suggestions can be made: In programming education, different programming tools can be used to detect their impact on students' achievement, problem-solving skills, and attitude towards the course. The effectiveness of the Scratch program in programming education can also be tested with

different age groups. Another investigation may be whether the Scratch program has its effect in imparting different skills in thinking (in information processing). It is considered that work repeated as a result of extending the working period could make a statistically significant difference in building program solving skills.

References

Akpınar, Y., & Altun, A. (2014). Bilgi toplumu okullarında programlama eğitimi gereksinimi [The Need for Programming Education in Information Society Schools]. *İlköğretim Online* [Elementary Education Online], *13* (1), 3. Retrieved from http://ilkogretim-online.org.tr

Alp, Y. (2019). *Blok tabanlı programlama öğretiminin ortaokul öğrencilerinin problem çözme becerisine ve bilgisayara yönelik tutumuna etkisi* [The effects of block-based programming on the problem solving ability and attitude towards computer for secondary school students] (Master's thesis, Inönü University, Malatya, Turkey). Retrieved from https://tez.yok.gov.tr/UlusalTezMerkezi/

Avcı, Ü., & Ersoy, H. (2018). Bilgisayar programlama derslerinde öğrenme motivasyonu ölçeğinin Türkçe uyarlaması: geçerlilik ve güvenilirlik çalışması. [The Adaptation of Learning Motivation in Computer Programming Courses Scale into Turkish: The Study of Validity and Reliability]. *Yükseköğretim ve Bilim Dergisi* [Journal of Higher Education & Science], *8* (1), 73-81. doi: 10.5961/jhes.2018.249

Coşar, M. (2013). *Problem temelli öğrenme ortamında bilgisayar programlama çalışmalarının akademik başarı, eleştirel düşünme eğilimi ve bilgisayara yönelik tutuma etkileri* [*Effects of computer programming studies on academic success, critical thinking skills and programming-based attitudes in problem-based learning environment*] (Doctoral dissertation, Gazi University, Ankara, Turkey). Retrieved from https://tez.yok.gov.tr/UlusalTezMerkezi/

Çam, E. (2019). *Robotik destekli programlama eğitiminin problem çözme becerisi, akademik başarı ve motivasyona etkisi* [*The influence of robotic assisted programming education on problem solving skills, academic success and motivation*] (Doctoral dissertation, Sakarya University, Sakarya, Turkey). Retrieved from https://tez.yok.gov.tr/UlusalTezMerkezi/

Çatlak, Ş., Tekdal, M., & Baz, F. Ç. (2015). Scratch yazılımı ile programlama öğretiminin durumu: bir doküman inceleme çalışması. [The Status of Teaching Programming with Scratch: A Document Review Work]. *Journal of Instructional Technologies & Teacher Education, 4*(3), 13-25.

Çetin, E. (2012). *Bilgisayar programlama eğitiminin çocukların problem çözme becerileri üzerine etkisi* [The effect of computer programming education on children's problem-solving skills] (Master's thesis, Gazi University, Ankara, Turkey). Retrieved from https://tez.yok.gov.tr/UlusalTezMerkezi/

Dinçer, A. (2018). *6.sınıf öğrencilerine Scratch ve Kodu game lab programlama dillerinin öğretiminde öğrencilerin tutum, öz yeterlilik ve akademik başarılarının karşılaştırılması* [The comparison of 6th grade students' in terms of attitudes, self-efficacy and academic achievement on teaching of scratch and kodu game lab 'programming languages] (Master's thesis, Dokuz Eylül University, İzmir, Turkey). Retrieved from https://tez.yok.gov.tr/UlusalTezMerkezi/

Erol, O. (2015). *Scratch ile programlama öğretiminin bilişim teknolojileri öğretmen adaylarının motivasyon ve başarılarına etkisi.* [*The effects of teaching programming with Scratch on pre-service information technology teachers' motivation and achievements*] (Doctoral dissertation, Anadolu University, Eskişehir, Turkey). Retrieved from https://tez.yok.gov.tr/UlusalTezMerkezi/

Ersoy, H., Madran, R. O., & Gülbahar, Y. (2011, February*). Programlama dilleri öğretimine bir model önerisi: robot programlama.* [A Model Proposed for Teaching Programming Languages: Robotic Programming] Akademik Bilişim, 11. Malatya, Turkey.

Genç, Z., & Tınmaz, H. (2010). *Programlama dilleri dersi içerisinde öğrencilerin problem çözme becerileri ve bilgisayar tutumları değerlendirmesi: Bir Fırat Üniversitesi örneği*. [Evaluation of Students' Problem Solving Skills and Attitude Toward Computers in a Programming Course: A Case Study From Fırat University] In International Education Technology Conference, *IETC*. İstanbul, Turkey.

Genç, Z., & Karakuş, S. (2011).*Tasarımla öğrenme: eğitsel bilgisayar oyunları tasarımında Scratch kullanımı*. In 5th International Computer & Instructional Technologies Symposium (ICITS), Elazığ, Turkey.

ISTE (International Society for Technology in Education) Standards For Students, 2016 (http://www.iste.org/standards/for-students#startstandards).

Işık, A. D. (2010). *Bilişim teknolojileri dersi için oluşturmacı yaklaşım doğrultusunda hazırlanan öğrenme paketinin etkileri* [*The effects of learning package prepared according to the constructivist approach on information technologies course*] (Doctoral dissertation, Dokuz Eylül University, İzmir, Turkey). Retrieved from https://tez.yok.gov.tr/UlusalTezMerkezi/

Kalelioğlu, F., & Gülbahar, Y. (2014). The effects of teaching programming via Scratch on problem-solving skills: a discussion from learners' perspective. Informatics in Education. *Informatics in* Education, *13* (1), 33–50. Retrieved from https://www.mii.lt/informatics_in_education/pdf/infe232.pdf

Kardaş, N., Anagün, Ş. S., & Yalçınoğlu, P. (2014). Problem çözme envanterini ilköğretim öğrencilerine uyarlama çalışması: doğrulayıcı faktör analizi sonuçları [Adaptation of problem solving inventory for the elemantary school students: confirmatory factor analysis]. *Elektronik Sosyal Bilimler Dergisi* [Electronic Journal of Social Sciences] *13* (51), 182-194. Retrieved from http://dergipark.gov.tr/download/article-file/70583

Korkmaz, Ö. (2016). The effects of Scratch-based game activities on students' attitudes, self-efficacy, and academic achievement. *International Journal of Modern Education & Computer Science, 8* (1), 16-23. DOI: 10.5815/ijmecs.2016.01.03

Korkmaz, Ö. (2018). The effect of Scratch- and Lego Mindstorms Ev3-Based programming activities on academic achievement, problem-solving skills, and logical-mathematical thinking skills of students. *MOJES: Malaysian Online Journal of Educational Sciences, 4* (3), 73-88. Retrieved from https://mojes.um.edu.my/article/view/12658

Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The Scratch programming language and environment. *ACM Transactions on Computing Education, 10* (4), 16. http://doi.acm.org/10.1145/1868358.1868363.

Miles, M. B., & Huberman, A. M. (1984). Qualitative Data Analysis: An expanded sourcebook. Sage.

Ministry of National Education. (2012). *Ortaokul ve imam hatip ortaokulu bilişim teknolojileri ve yazılım dersi öğretim programı* [Secondary school and imam hatip

secondary school information technologies and software course teaching program]. Ankara, Turkey: Talim Terbiye Kurulu Başkanlığı.

Ministry of National Education. Milli Eğitim Bakanlığı. (2018). *Bilişim teknolojileri ve yazılım dersi öğretim programı.* [Information technologies and software course teaching program]. Ankara, Turkey: Talim Terbiye Kurulu Başkanlığı.

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,* 723-727, Putrajaya, Malaysia.

Ouahbi, I., Kaddari, F., Darhmaoui, H., Elachqar, A., & Lahmine, S. (2015). Learning basic programming concepts by creating games with a Scratch programming environment. *Procedia-Social and Behavioral Sciences*, 191, 1479-1482. DOI: 10.1016/j.sbspro.2015.04.224

Ozoran, D., Cagiltay, N., & Topalli, D. (2012). *Using Scratch in the introduction to programming course for engineering students.* In 2nd International Engineering Education Conference, 125-132.

Özer, F. (2019). *Kodlama eğitiminde robot kullanımının ortaokul öğrencilerinin erişi, motivasyon ve problem çözme becerilerine etkisi* [Effect of using robotics in teaching coding on achievement, motivation and problem solving skills of middle school students] (Master's thesis, Hacettepe University, Ankara, Turkey). Retrieved from https://tez.yok.gov.tr/UlusalTezMerkezi/

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

Sayın, Z. (2017). Bilgisayar bilimi eğitimi kapsamı [Computer science education scope]. Y. Gülbahar (Ed.), B*ilgi işlemsel düşünmeden programlamaya* [From computational thinking to programming] (pp. 13-141). Ankara, Turkey: Pegem Akademi.

Sönmez, V., & Alacapınar, F. G. (2014). *Örneklendirilmiş bilimsel araştırma yöntemleri* [Sampled scientific research methods] (3. Baskı) Ankara, Turkey: Anı Yayıncılık.

Şahin, Ç. (2004). Problem çözme becerisinin temel felsefesi. *Kazım Karabekir Eğitim Fakültesi Dergisi*, 10.

Şimşek, İ. (2018). Dünyada programlama öğretimi [Programming teaching in the world]. In Y. Gülbahar ve H. Karal (Ed.), *Kuramdan uygulamaya programlama öğretimi [Teaching programming from theory to practice]* (pp. 38-65). Ankara, Turkey: Pegem Akademi.

Utting, I., Cooper, S., Kölling, M., Maloney, J., & Resnick, M. (2010). Alice, Greenfoot, and Scratch--a discussion. *ACM Transactions on Computing Education, 10* (4), 17. https://doi.org/10.1145/1868358.1868364

Yükseltürk, E,. & Altıok, S. (2015). Bilişim teknolojileri öğretmen adaylarının bilgisayar programlama öğretimine yönelik görüşleri [Pre-Service Information Technologies Teachers' Views on Computer Programming Teaching]. *Amasya Üniversitesi Eğitim Fakültesi Dergisi* [Amasya Education Journal], *4* (1), 50-65.

Yükseltürk, E., & Altıok, S. (2016). Bilişim teknolojileri öğretmen adaylarının programlama öğretiminde Scratch aracının kullanımına ilişkin algıları [Pre-Service Information Technology Teachers` Perceptions about Using Scratch Tool in Teaching Programming]. *Mersin Üniversitesi Eğitim Fakültesi Dergisi* [Mersin University Journal of the Faculty of Education], *12* (1), 39-52.