Ilkogretim Online - Elementary Education Online, 2020; 19 (4): pp. 2171-2182
http://ilkogretim-online.org.tr
doi:10.17051/ilkonline.2020.763834

# Effects of robotic coding activities on the effectiveness of secondary school students' self-efficacy for coding*

**İbrahim Kasalak,** *Yenimahalle Science and Art Center,* Turkey, *i.kasalak@gmail.com* ORCID: 0000-0003-3824-7351
**Arif Altun,** *Hacettepe University,* Turkey *altunar@gmail.com* ORCID: 0000-0003-4060-6157

**Abstract.** In this study, it was aimed to determine whether robotic coding activities at secondary school level had any significant effect on students' self-efficacy perceptions related to block-based programming. Data were collected by "self-proficiency perception scale for block-based programming" prepared by Altun and Kasalak (2018). Within the scope of the study, 5-week robotic coding activities were planned, followed by 58 students in a public school. According to pre-test and post-test results of the students, there was a significant change in intra-group positive direction in the self-efficacy perception scores of both simple and complex block-based programming ($t_{simple}$ = -5.01, p = 0.00, $t_{complex}$ = -8.84, p = 0.00). Across various variables, when we look at the differentiation of self-efficacy perceptions regarding block-based programming, it is found that it does not differ significantly according to gender ($t_{simple}$ = -0.58, p = 0.56, $t_{complex}$ = 0.87, p = 0.39), computer ownership at home ($t_{simple}$ = -1.23, p = 0.22, $t_{complex}$ = -1.23, p = 0.22), Internet connection ownership at home ($t_{simple}$ = -0.37, p = 0.22, $t_{complex}$ = -0.44, p = 0.66) and the possibility to study Scratch program out of course ($t_{simple}$ = -0.91, p = 0.37, $t_{complex}$ = -0.91, p = 0.37).

**Keywords:** Blok based programming, coding, robotic coding, self-efficacy perception

Received: 20.03.2019 Accepted: 14.12.2019 Published: 15.09.2020

## INTRODUCTION

According to Resnick et al. (2009), the new generation youth can easily send messages, play online games and browse the internet. However, only a few of the young people who always interact so much with digital media can do their own games, animations and simulations. In other words, they can "read" but not "write". According to Resnick et al. (2009), digital fluency requires not only chat (today's use of social media), research and interact on the Internet, but to the ability to design, create and invent.

As Resnick et al. (2009) coined the term "digital fluency", it gained more importance recently with the incredible evolution of technology; and, education as a sector have started to pay more attention in line with the future predictions of economists and the demands of the industry sector, on developing education systems for training individuals with 21st century skills. Examples of these are STEM (Science (S), Technology(T), Engineering(E) and Mathematics(M)), which embodies education initiatives with real interdisciplinary concrete practical activities to find solutions to daily life problems (in different disciplines, coding, robotic coding and smart device design activity examples are encountered in various sizes), initiatives for coding education and robotic coding education, and the radical change in the Finnish education system at the end of 2016.

When the studies carried out within the scope of programming education are analyzed, the studies in which programming education was generally handled as software expertise in previous years are reached. In some studies carried out at A-6 level, it was concluded that programming education could not be given at this age due to the students' inadequate abstraction skills (e.g. Armoni, 2012). However, with the emergence of block-based examples of programming, especially in studies conducted in the last ten - fifteen years; there are findings that block-based programming can also yield positive results at A-12, A-8 and A-6 levels, students are eager to participate in activities, are interested in, and a positive increase in programming performance,

* This study was a part of first author's master thesis entitled "Effects of robotic coding activities on the effectiveness of secondary school students' self-efficacy and student experience about activities" under the supervision of the second author.

problem solving and computational thinking skills is achieved (Resnick et al., 2009; Kalelioğlu & Gülbahar, 2014; Kukul & Gökçearslan, 2014; Yükseltürk & Altıok, 2016).

Computational thinking and programming skills are considered among the skills that individuals should have in the 21st century society (ISTE, 2016; Angeli et al., 2016). While programming does not mean computational thinking, many researchers and scientists working on this subject think that programming education contributes to individuals' computational thinking skills (Wing, 2006; Settle & Perkovic, 2010; Gülbahar & Kalelioğlu, 2014; Allsop, 2015; Angeli et al., 2016). Programming education, with its positive contribution to computational thinking skills, has also started to be handled within the scope of basic skills education apart from specialization education. Akpınar and Altun (2014) emphasize that the concepts such as algorithm, programming and database should be accepted as the knowledge and skills that should be gained within the scope of basic education starting from 5th grade of primary education.

The fact that programming is done on a block-based platform has made it possible to train even at the secondary level, causing coding education to be included among training programs at A-8 and A-12 levels in many countries (Code.org 2017 Annual Report, 2018). In parallel with the studies carried out within the scope of computational thinking and programming education in the world, programming education in Turkey has been included in the Curriculum (TTKB, 2012; TTKB, 2016; TTKB, 2017). In the computer science curriculum for high schools, programming and topics related to robotics programming are included (TTKB, 2016). In the renewed information technologies and software curriculum, 50% of the curriculum at the 5th and 6th grades are devoted to the “problem solving and programming” unit (TTKB, 2017).

In recent years, there are examples in the literature that use the term "coding" instead of "programming". To make an analogy, coding can be seen not as the literacy of the programming process, but as the reading - writing process of programming. Being a programming literate will involve a process that will mean more than coding activities.

Programming activities in block-based programming languages such as Scratch, Blockly, Alice, designed to fulfill predefined tasks are also called coding, and publications where block-based programming term and coding expression are used interchangeably are also found in the literature (e.g. Resnick et al., 2009). Although the coding term is a small stage of the software development process, it has become a slogan used to promote programming activities organized at A-12 level for reasons such as computational thinking, 21st century skills acquisition, and the skills needed in the industry 4.0 process (eg, Coding week, coding activities, robotic coding, Kodla(Ma)nisa, KodlaRize, KodlaSakarya). Within the scope of this study, the use of block-based programming expression and coding expression in the same sense was preferred.

In the literature, robotic coding activities are also referred to as physical coding. Researchers have reported findings that students are more interested in activities involving real world objects rather than virtual objects, and that students have fun with physical coding and want to stay longer to continue the activities (Rusk, Resnick, Berg, & Pezalla-Granlund, 2008; Przybylla & Romeike, 2014). Robotic coding applications, embodies software processes that require an abstract process and allow students to directly observe how the codes they have written can work with a hardware after compiling. For this reason, many educators prefer to diversify their programming instruction with such hardware support. It can be said that it is very important to present our students the robotic coding activities that contain the coding activities known to contribute to the "computational thinking" and "programming" skills considered to be among the skills of the 21st century and that can be seen in concrete terms of internet of things and smart devices. However, there has not been enough research findings in the field of robotic coding activities at A-12, A-8 and A-6 levels.

In robotic coding activities, as in other programming activities, coding can be done with text-based and block-based programming languages. In the robotic coding activities to be held at the 5th grade of the secondary school, care was taken to choose a programming language suitable for the students' development levels. When the studies conducted in this context are examined, it has been seen that the most widely used block-based programming language in coding education is Scratch. Scratch is used in more than 40 languages in more than 150 countries today (https://scratch.mit.edu/about / 10.3.2018). The Scratch for Arduino program is a Scratch modification (Scratch for Arduino, 2018), which provides the Scratch program with new blocks to manage Arduino-connected sensors and actuators. The findings of coding activities using Scratch, a block-based programming language, are also expected to be similarly obtained in coding activities with Scratch for Arduino, a modification of Scratch. The screenshot and the motion block of the Scratch program are shown in Figure 1, and the screenshot and the motion block of the Scratch for Arduino program are shown in Figure 2. From the figures, it appears that the Scratch for Arduino program is a Scratch modification with new blocks added to the Scratch program's motion block to manage Arduino sensors and actuators.


**FIGURE 1:** *Screenshots of Scratch program*

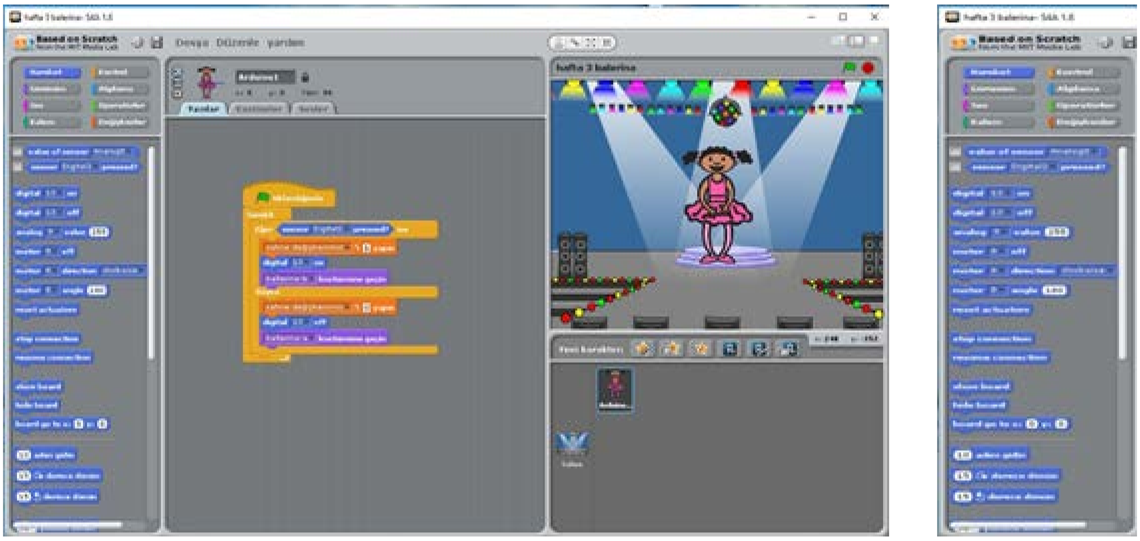

**FIGURE 2:** *Screenshots of Scratch for Arduino program*

According to Bandura (1977), individuals' judgment about their own success in performing certain tasks, that is, self-efficacy perception, is directly related to their performance and effort in performing that task successfully. It is stated that individuals with high self-efficacy perception are more resistant to cope with difficulties and will strive harder to reach goals; individuals with low self-efficacy perception tends to exaggerate problems; therefore, they can complicate the existing problem even more (Davidson, Larzon, & Ljunggren, 2010). Stajkovic and Luthans (1998) analyzed 114 studies dealing with the relationship between self-efficacy perception and performance, and concluded that there was a significant and positive relationship between self-efficacy perception and task performance. Therefore, high self-efficacy perception of individuals in any context is important as it will be directly related to its performance.

There are studies emphasizing the relationship between perceived self-efficacy and programming performance in the literature (Aşkar & Davenport, 2009; Altun & Mazman, 2013; Yükseltürk & Altıok, 2016; Mazman Akar & Altun, 2017). However, while robotic coding activities have become increasingly widespread both in formal education processes and out-of-school settings, not enough research have been reported regarding the effect of robotic coding activities on participants' coding self-efficacy perceptions. Therefore, in this study, it is aimed to observe whether students' self-efficacy perceptions related to block-based programming have changed after robotic coding activities. In other words, the main purpose of this study is to investigate whether robotic coding activities at secondary school level would have a significant effect on students' self-efficacy perceptions related to block-based programming.

## METHODS

This study is designed as single-group pretest-posttest experimental research to investigate the effects of robotic coding activities on secondary school students' self-efficacy perceptions related to block-based programming.

Since the effect of robotic coding activities on students' self-efficacy perceptions of coding will be investigated with robotic coding activities, a five-week Scratch coding training was organized by the researcher. After the five-week programming training with two-hour lessons per week with Scratch program, five-week robotic coding activity planning was conducted as two-hour lessons per week. After each activity, in line with the data obtained from the observations and interviews related to the related activity, the subjects where the students needed additional information and the difficulties in learning processes were determined. In line with the findings, corrections and arrangements were reflected on the next activity plan.

In addition to preparing the data collection tools and activity plans for the activity before the robotic coding activities, the procurement process of the activity tools and the necessary equipment was completed. During the activity, the robotic parts were placed in a box and turned into a training set so that the students could perform the activities regularly. As the activities will be held in the information technologies and software course opened within the scope of the "supporting and training course", classes are organized for a maximum of 20 people, as per the regulation. Therefore, 10 robotic sets were prepared considering that the students would work in pairs. The robotic coding training set prepared for use in robotic coding activities is shown in Figure 3:

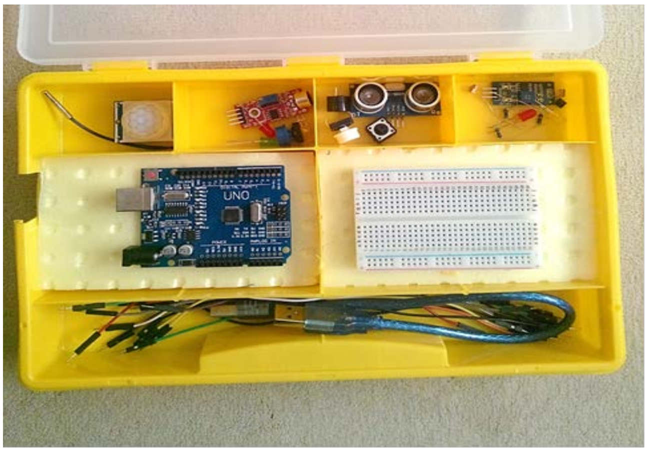


**FIGURE 3:** *Robotic coding training set prepared for activities*

**Study group**

The research group consisted of 66 students who followed the "supporting and training course" over the weekend. Although the number of students enrolled in the course was 74 at the beginning, 66 students attended the course. In addition, eight of these students dropped out due to health problems and/or school changes. Therefore, the research was completed with the participation of 58 students. Descriptive statistics obtained from the demographic data of the research group are given in Table 1:

**Table 1.** *Descriptive data of the robotic coding education study group*

| | | N | % |
|---|---|---|---|
| Gender | Female | 27 | 46.6 |
| | Male | 31 | 53.4 |
| Grade level | 5th grade | 58 | 100 |
| Computer ownership at home | Yes | 34 | 58.6 |
| | No | 24 | 41.4 |
| Internet connection ownership at home | Yes | 24 | 41.4 |
| | No | 34 | 58.6 |
| Having previously taken a Scratch program course | Yes | 58 | 100 |
| | No | 0 | 0 |
| Still taking a Scratch program course | Yes | 0 | 0 |
| | No | 58 | 100 |
| Having previously taken a Scratch for Arduino robotic course | Yes | 0 | 0 |
| | No | 58 | 100 |
| The possibility to study Scratch program out of course | Yes | 20 | 34.5 |
| | No | 38 | 65.5 |
| Frequency of study to Scratch program out of course | 1 day a week | 6 | 10.3 |
| | 1-2 days a week | 5 | 8.6 |
| | 3-4 days a week | 4 | 6.9 |
| | More than 1 hour every day | 1 | 1.7 |
| | Less than 1 hour every day | 4 | 6.9 |
| The possibility to study Scratch for Arduino program out of course | Yes | 0 | 0 |
| | No | 58 | 100 |

## Design of the Teaching Process

In order to ensure that students in the research group have prior knowledge of coding, a five-week Scratch training was held before the research process. After the coding activities were completed with Scratch, robotic coding activities were organized.

The process of designing robotic coding activities and preparing activity plans were designed in two phases. The first phase of the study was the process of determining the goals of the activities and designing the activities required to achieve these goals. In this context, learning objectives were determined by two academics from Hacettepe University Computer Education and Instructional Technologies Education Department, and the researcher and an another computer teacher, taking into account the programming features in Scratch, which was specified in an article prepared by Scratch's developer Massachusetts Institute of Technology (MIT) Media Lab's Lifelong Kindergarten Group (Resnick et al., 2009). While determining the objectives, care was taken to ensure that the electronic circuit tasks were at a level that the students could overcome.

The second phase of the study included the determination of the prior knowledge and skills required for this activity of the students and the preliminary studies required to meet these prior knowledge and skills. Although these two phases of the study may seem like two separate studies, the objectives and activity process for coding cannot be determined without considering the complexity level that students can design their electronic circuits since the robotic coding activities included microcontroller cards such as Arduino Uno R3. When the tasks in robotic coding activities are examined, it is envisaged that some knowledge and skills related to electricity and electronics must have been provided to students in order to fulfil these tasks. Therefore, a preliminary task analysis was carried out with two science teachers about the level of secondary school 5th grade students in the target group. As a result of the interdisciplinary preparation, it was observed that these tasks are included in the 5th grade science curriculum as well as in the 6th, 7th and 8th grades. As the objectives of the activities are not related to science and technology, no reporting has been made for these achievements here.

**FIGURE 4:** *Students during the robotic coding activity*

As a result of the preliminary studies, a 5-week activity plan was prepared, and the corrections and changes were made in the plans for the next activity, taking into account the observation findings obtained during the activities.

## Data Collection Tools

“The self-efficacy perception scale related to block-based programming” scale, prepared by Altun and Kasalak (2018), was utilized in this study. While the first part of the scale consists of demographic data about students, the second part consists of self-efficacy perception items related to block-based programming. The scale had two constructs: first one is named as "simple block-based programming tasks" and consisted of five items; and, the second factor is named as “complex block-based programming tasks" and had seven items. The internal consistency coefficient of the scale (Cronbach alpha) is .893, and the scale explained 58.225% of the total variance. The maximum score that can be obtained for self-efficacy perception related to simple block-based programming is 25, the minimum score is 5; the maximum score for self-efficacy perception related to complex block-based programming is 35, and the minimum score is 5 points.

## Data analysis

Demographic data was analyzed with descriptive statistics. Scale data was analyzed with paired samples t-test to explore the intra-group changes in students' self-efficacy perception scores related to block-based programming. P value is set to be 0.05 in those analyses.

# RESULTS

In this section, the research findings and results of the analyses will be presented in the light of each sub-question. First, descriptive statistics and preliminary analysis will be presented under preliminary findings, then research questions will be presented separately.

## Preliminary Analysis

At the end of the robotic block-based programming training, the differences in self-efficacy perception scores across various variables (gender, computer ownership at home, Internet connection ownership at home, the possibility to study Scratch program out of course) were examined. Prior to this, analyzes were needed on the readiness levels of individuals to determine whether there was a significant difference in the self-efficacy perception scores across various variables.

Before the activities were realized, an independent samples t-test was conducted to reveal whether self-efficacy perception pretest scores differ in simple and complex block-based programming scores across various variables (gender, computer ownership at home, Internet connection ownership at home, the possibility to study Scratch program out of course). In Table 2, independent sample t-test results are presented.

According to independent samples t-test results, individuals' self-efficacy perceptions related to both simple and complex block-based programming did not differ significantly in pre-test scores according to gender ($t_{simple} = 0.17$, $p = 0.86$, $t_{complex} = -0.75$, $p = 0.46$), computer ownership at home ( $t_{simple} = 1.36$, $p = 0.18$, $t_{complex} = 0.95$, $p = 0.35$), internet connection ownership at home ($t_{simple} = 0.39$, $p = 0.70$, $t_{complex} = -0.32$, $p = 0.75$) and the possibility to study Scratch program out of course ($t_{simple} = 1.21$, $p = 0.23$, $t_{complex} = 0.37$, $p = 0.72$).

**Table 2.** *Self-efficacy perception pretest scores related to block-based programming across various variables*

| | | | N | $\bar{X}$ | SD | t | df | p | Mean Difference | 95% Confidence Interval of the Difference Lower | Upper |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Gender | Female | Simple | 27 | 21.89 | 3.07 | 0.17 | 56 | 0.86 | 0.14 | -1.49 | 1.77 |
| | Male | | 31 | 21.75 | 3.12 | | | | | | |
| | Female | Complex | 27 | 25.87 | 4.95 | -0.75 | 56 | 0.46 | -0.95 | -3.50 | 1.60 |
| | Male | | 31 | 26.82 | 4.73 | | | | | | |
| Computer ownership at home | Yes | Simple | 34 | 22.27 | 2.85 | 1.36 | 56 | 0.18 | 1.10 | -0.52 | 2.73 |
| | No | | 24 | 21.17 | 3.31 | | | | | | |
| | Yes | Complex | 34 | 26.89 | 4.78 | 0.95 | 56 | 0.35 | 1.22 | -1.36 | 3.79 |
| | No | | 24 | 25.67 | 4.88 | | | | | | |
| Internet connection ownership at home | Yes | Simple | 24 | 22.00 | 2.83 | 0.39 | 56 | 0.70 | 0.32 | -1.33 | 1.97 |
| | No | | 34 | 21.68 | 2.26 | | | | | | |
| | Yes | Complex | 24 | 26.14 | 4.42 | -0.32 | 56 | 0.75 | -0.41 | -3.00 | 2.18 |
| | No | | 34 | 26.55 | 5.14 | | | | | | |
| The possibility to study Scratch program out of course | Yes | Simple | 20 | 22,48 | 1,98 | 1.21 | 56 | 0.23 | 1.02 | -0.67 | 2.71 |
| | No | | 38 | 21,46 | 3,48 | | | | | | |
| | Yes | Complex | 20 | 26,70 | 5.09 | 0.37 | 56 | 0.72 | 0.49 | -2.20 | 3.18 |
| | No | | 38 | 26.21 | 4.73 | | | | | | |

## Robotic coding activities and self-efficacy perception for coding

At the end of the robotic coding activities, paired samples t-test was performed to analyze the intra-group changes of students' self-efficacy perception scores related to block-based programming. In Table 3, there are intra-group differentiation findings of self-efficacy perceptions related to simple block-based programming and self-efficacy perceptions related to complex block-based programming at the end of the robotic coding training.

Intra-group changes of students' self-efficacy perception scores regarding block-based programming are presented in Table 3. According to pre-test and post-test results, there was a significant difference in intra-group positive direction in the self-efficacy perception scores of both simple and complex block-based programming ($t_{simple}$ = -5.01, p = 0.00; $t_{complex}$ = -8.84, p = 0.00).

**Table 3.** *Self-efficacy perceptions of coding at the end of robotic coding training intra-group differentiation table*

| | | | $\bar{X}$ | SD | t | df | p |
|---|---|---|---|---|---|---|---|
| Intra-Group Change | Simple | Pre-Test | 21.81 | 3.07 | -5.01 | 57 | 0.00 |
| | | End-Test | 24.10 | 1.62 | | | |
| | Complex | Pre-Test | 26.38 | 4.82 | -8.84 | 57 | 0.00 |
| | | End-Test | 32.26 | 3.23 | | | |

## Robotic coding activities and self-efficacy perception related to block-based programming across various variables

At the end of the training, an independent samples t-test was performed to analyze the changes in students' self-efficacy perception scores related to block-based programming across various variables. In Table 4, results are presented.

**Table 4.** *Analysis table of self-efficacy perceptions related to coding at the end of robotic coding training according to various variables in intra-group*

| | | | | | | | | | | 95% Confidence Interval of the Difference | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | N | $\bar{X}$ | SD | t | df | p | Mean Difference | Lower | Mean Difference |
| Gender | Female | Simple | 27 | 2.00 | 3.45 | -0.58 | 56.00 | 0.56 | -0.54 | -2.38 | 1.31 |
| | Male | | 31 | 2.54 | 3.55 | | | | | | |
| | Female | Complex | 27 | 6.50 | 5.54 | 0.87 | 56.00 | 0.39 | 1.16 | -1.51 | 3.83 |
| | Male | | 31 | 5.34 | 4.63 | | | | | | |
| Computer ownership at home | Yes | Simple | 34 | 1.82 | 3.07 | -1.23 | 56.00 | 0.22 | -1.13 | -2.98 | 0.72 |
| | No | | 24 | 2.95 | 3.96 | | | | | | |
| | Yes | Complex | 34 | 5.23 | 4.74 | -1.16 | 56.00 | 0.25 | -1.56 | -4.25 | 1.13 |
| | No | | 24 | 6.79 | 5.45 | | | | | | |
| Internet connection ownership at home | Yes | Simple | 24 | 2.08 | 3.33 | -0.37 | 56.00 | 0.71 | -0.35 | -2.22 | 1.52 |
| | No | | 34 | 2.43 | 3.62 | | | | | | |
| | Yes | Complex | 24 | 5.53 | 4.66 | -0.44 | 56.00 | 0.66 | -0.60 | -3.32 | 2.12 |
| | No | | 34 | 6.13 | 5.38 | | | | | | |
| The possibility to study Scratch program out of course | Yes | Simple | 20 | 1.72 | 2.42 | -0.91 | 56.00 | 0.37 | -0.87 | -2.80 | 1.06 |
| | Yes | | 38 | 2.59 | 3.92 | | | | | | |
| | Yes | Complex | 20 | 6.05 | 5.06 | 0.19 | 56.00 | 0.85 | 0.26 | -2.56 | 3.09 |
| | No | | 38 | 5.79 | 5.12 | | | | | | |

## Gender and coding self-efficacy perception

Firstly, at the end of robotic block-based programming training, independent samples t-test was conducted to reveal whether students' self-efficacy perception scores related to simple block-based programming and self-efficacy perception scores related to complex block-based programming differ according to gender. According to independent samples t-test results, it was found that individuals' self-efficacy perception scores related to simple block-based programming ($t_{simple}$ = -0.58, p = 0.56) and self-efficacy perception scores related to complex block-based programming ($t_{complex}$ = 0.87, p = 0.39) did not differ according to gender.

**Computer ownership at home and coding self-efficacy perception**

At the end of robotic block-based programming training, independent samples t-test was conducted to reveal whether students' self-efficacy perception scores related to simple block-based programming and self-efficacy perception scores related to complex block-based programming differ according to computer ownership at home. According to independent samples t-test results, it was found that individuals' self-efficacy perception scores related to simple block-based programming ($t_{simple}$ = -1.23, p = 0.22) and self-efficacy perception scores related to complex block-based programming ($t_{complex}$ = -1.16, p = 0.25) did not differ according to computer ownership at home.

**Internet connection ownership at home and coding self-efficacy perception**

At the end of robotic block-based programming training, independent samples t-test was conducted to reveal whether students' self-efficacy perception scores related to simple block-based programming and self-efficacy perception scores related to complex block-based programming differ according to Internet connection ownership at home. According to independent samples t-test results, it was found that individuals' self-efficacy perception scores related to simple block-based programming ($t_{simple}$ = -0.37, p = 0.71) and self-efficacy perception scores related to complex block-based programming ($t_{complex}$ = -0.44, p = 0.66) did not differ according to Internet connection ownership at home.

**The possibility to study Scratch program out of course and coding self-efficacy perception**

At the end of robotic block-based programming training, independent samples t-test was conducted to reveal whether students' self-efficacy perception scores for simple block-based programming and self-efficacy perception scores for complex block-based programming differ according to the possibility to study Scratch program out of course. According to independent samples t-test results, it was found that individuals' self-efficacy perception scores related to simple block-based programming ($t_{simple}$ = -0.91, p = 0.37) and self-efficacy perception scores related to complex block-based programming ($t_{complex}$ = 0.19, p = 0.85) did not differ according to the possibility to study Scratch program out of course.

## DISCUSSION and CONCLUSIONS

After the robotic coding activities, according to pre-test and post-test paired samples t-test results of the students, there was a significant change in the self-efficacy perception scores of both simple and complex block-based programming ($t_{simple}$ = -5.01, p = 0.00; $t_{complex}$ = -8.84, p = 0.00) in positive direction. While this finding supports the findings of the study by Rusk, Resnick, Berg, and Pezalla-Granlund (2008), it does not match to the study conducted by Beug (2012). Within the scope of this study, paying attention to organizing activity plans in accordance with the principles stated in the studies of Rusk, Resnick, Berg, and Pezalla-Granlund (2008) that “students should not have any hardware problems during the activities, events should be organized to appeal to the students' interests”, can be shown as the reason for the results of the two studies to support each other. In the study of Beug (2012), the block-based programming language Scratch and the text-based programming language Arduino were compared. Furthermore, the environment was not prepared during the activities, and problems such as introducing the Arduino card to computers could be said as the reasons for the robotic coding activities to have a negative effect on student performance.

Regarding the validity and reliability of these findings, an initial analysis was run to observe whether there was any significant difference between participants according to various variables such as computer ownership at home, since they could cause a significant change in self-efficacy perceptions. According to the pre-test data analysis findings, the self-efficacy perception test scores of the individuals related to both simple and complex block-based programming were found that no significant difference was observed according to computer ownership at home, Internet connection ownership at home and the possibility to study Scratch program out of course. The findings show that there is no significant difference in the pre-test scores of the students in the research group according to various variables. According to these findings, it can be concluded that participants' readiness levels were similar and did not have a detrimental effect to lead further changes in the self-efficacy perceptions related to block-based programming.

At the end of robotic coding activities, independent samples t-test was performed in order to analyze the intra-group changes according to various variables in self-efficacy perceptions related to block-based programming. The findings show that there is no significant difference in students' self-efficacy perceptions related to block-based programming at the end of robotic coding activities according to various variables. While designing robotic coding activities, efforts were made to create an activity environment in accordance with the determination of "Care should be taken to create groups according to the student' interests, and activities should be created in accordance with the students' interests" stated by Rusk, Resnick, Berg and Pezalla-Granlund (2008). It can be said that this situation arising from the activity design process is effective in the fact that both genders get more efficiency from the activities and that there is no difference due to gender in the change in self-efficacy perceptions.

The fact that none of the students have access to robotic sets out of course can be shown as the reason that they do not have access to robotic coding applications out of course and do not make a meaningful change in self-efficacy perceptions of these variables even if they have computer ownership at home and Internet ownership connection at home. If students have access to robotic sets out of the course, the results may differ. Considering the effects of this differentiation in groups, students' self-efficacy perceptions and changes in their task performance can be investigated. Within the scope of this study, the participants were expected to place the necessary parts on the electronic circuit and then make the coding in order to perform the tasks. However, in the scope of the study, no research questions related to the electronic circuit implementation has been included. In later studies, research questions regarding interdisciplinary acquisitions, including the acquisitions of electricity and electronics, can be included. In these days, Arduino sets with electronic circuit elements pre-assembled are also becoming more common. In following studies, activities can be organized in such a way that they do not include the circuit-building tasks. Thus, it can be investigated whether students are able to focus more on coding processes. The limitation of STEM gains in such activities, which will not involve electronic circuit building tasks, can also be considered as another venue of discussion.

## REFERENCES


Akpınar, Y., & Altun, A. (2014). Bilgi toplumu okullarında programlama eğitimi gereksinimi. *Elementary Education Online, 13(1)*, 1-4.

Allsop, Y. (2015). Ict'den kodlamaya: İngiltere'de teknoloji eğitimi. *Eğitim Teknolojileri Zirvesi*, (s. 303-308). Ankara.

Altun, A., & Kasalak, İ. (2018). Blok temelli programlamaya ilişkin öz-yeterlik algısı geliştirme çalışması: Scratch örneği. *Eğitim Teknolojisi Kuram ve Uygulama, 8*(1), 209-225. doi:10.17943/etku.335916

Altun, A., & Mazman, S. G. (2013). Programlama-1 dersinin BÖTE bölümü öğrencilerinin programlamaya ilişkin öz yeterlilik algıları üzerine etkisi. *Journal of Instructional Technologies & Teacher Education, 2*(3), 24-29.

Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 computational thinking. *Educational Technology & Society, 19*(3), 47-57.

Armoni, M. (2012). Teaching CS in kindergarten: How early can the pipeline begin? *ACM Inroads, 3*(4), 18-19.

Aşkar, P., & Davenport, D. (2009). An investigation of factors related to self-efficacy for java programming. *The Turkish Online Journal of Educational Technology – TOJET January, 8*(1).

Bandura, A. (1977). Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review, 84*(2), 191-215.Beug, A. (2012). *Teaching introductory programming concepts: A comparison of Scratch and Arduino.* (Yayımlanmamış Yüksek Lisans Tezi). The Faculty of California Polytechnic State University, Obispo, San Luis.
Code.org 2017 Annual Report. (2018). *Global computer science education.* Mart 10, 2018 tarihinde https://goo.gl/cPCBgP adresinden alındı
Davidson, K., Larzon, L. & Ljunggren, K. (2010). *Self-Efficacy in programming among STS students. Technical Reports from Computer Science Education course of Upssala University. Ağustos 28, 2016 tarihinde https://goo.gl/52RbkD adresinden alındı*Gülbahar, Y., & Kalelioğlu, F. (2014). The effects of teaching programming. *Informatics in Education-An International Journal, 13*(1), 33-50.
ISTE. (2016). *ISTE Standarts for students 2016.* Aralık 19, 2016 tarihinde https://goo.gl/Z1E5bf adresinden alındı
Kasalak, İ. (2017). *Robotik kodlama etkinliklerinin ortaokul öğrencilerinin kodlamaya ilişkin öz-yeterlik algılarına etkisi ve etkinliklere ilişkin öğrenci yaşantıları.* (Yayımlanmamış Yüksek Lisans Tezi). Hacettepe Üniversitesi, Eğitim Bilimleri Enstitüsü, Ankara.
Kukul, V., & Gökçearslan, Ş. (2014). Scratch ile programlama eğitimi alan öğrencilerin problem çözme becerilerinin incelenmesi. *8. Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu*, (s. 58-63). Edirne.
Mazman Akar, S. G., & Altun, A. (2017). Individual differences in learning computer programming: A social cognitive approach. *Contemporary Educational Technology, 8*(3), s. 195-213.
Prensky, M. (2001, 9). Digital natives, digital immigrants. *On The Horizon, 9*(5), 1-6.
Przybylla, M., & Romeike, R. (2014). *Overcoming ıssues with students' perceptions of ınformatics in everyday life and education with physical computing - suggestions for the enrichment of computer science classes* (s. 6-20). Local Proceedings of the 7th International Conference on Informatics in; Situation, Evolution and Perspectives ISSEP 2014.
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).
Rusk, N., Resnick, M., Berg, R., & Pezalla-Granlund, M. (2008). New pathways into robotics: Strategies for broadening participation. *Journal of Science Education and Technology, 17*, 59-69.
Scratch for Arduino. (2018). *About S4A.* Mart 10, 2018 tarihinde http://s4a.cat adresinden alındı
Settle, A., & Perkovic, L. (2010). *Computational thinking across the curriculum: A conceptual framework.* Technical Reports. Aralık 23, 2016 tarihinde http://via.library.depaul.edu/tr/13 adresinden alındı
Stajkovic, A. D., & Luthans, F. (1998). Self-efficacy and work-related performance: A metaanalysis. *Psychological bulletin, 124*(2).TTKB. (2012). *Bilişim teknolojileri ve yazılım dersi ( 5, 6, 7 ve 8. sınıflar) öğretim programı.* Ankara: MEB.
TTKB. (2016). *Ortaöğretim bilgisayar bilimi dersi (kur 1, kur 2) öğretim programı.* Ankara: MEB.
TTKB. (2017). *Bilişim teknolojileri ve yazılım dersi öğretim programı.* Ankara: MEB.
Wing, J. M. (2006). Computational thinking. *Communications of The Acm, 49*(3), 33-35.
Yükseltürk, E., & Altıok, S. (2016). An investigation of the effects of programming with scratch. *British Journal of Educational Technology*. doi:10.1111/bjet.12453
Yükseltürk, E., & Altıok, S. (2016). Investigation of pre-service information technology teachers' game projects prepared with Scratch. *SDU International Journal of Educational Studies, 3*(1), 59-66.