Research Article / **Araştırma Makalesi**

Bolu Abant İzzet Baysal Üniversitesi

Eğitim Fakültesi Dergisi (BAİBÜEFD)

Bolu Abant Izzet Baysal University

Journal of Faculty of Education

2025, 25(1), 425– 445. https://dx.doi.org/ 10.17240/aibuefd.2025..-1443206

Scratch's Impact on Technological Pedagogical Content Knowledge, Computational Thinking Skills, and Computing Attitudes of Prospective Elementary Mathematics Teachers*

Scratch'in İlköğretim Matematik Öğretmen Adaylarının Teknolojik Pedagojik Alan Bilgileri, Bilgi İşlemsel Düşünme Becerileri ve Bilgisayar Bilimi Tutumları Üzerine Etkisi

Selva Büşra TURAN[1] , **Ahmet ERDOĞAN[2]**

Geliş Tarihi (*Received*): 26.02.2024 **Kabul Tarihi (*Accepted*):** 19.12.2024 **Yayın Tarihi (*Published*):** 15.03.2025

Abstract: The purpose of this study is to examine the impact of the Scratch program on prospective elementary mathematics teachers Technological Pedagogical Content Knowledge (TPACK), Computational Thinking Skills, and Computing Attitudes. The study involved 73 prospective elementary mathematics teachers. Among various experimental models, a single-group pretest-posttest experimental design was chosen for this study. Before and after the Scratch training, the Technological Pedagogical Content Knowledge (TPACK) Scale, the Computational Thinking Skills Scale and the Computing Attitude Scale were administered to the prospective teachers. The Wilcoxon signed-rank test and the dependent groups t-test were utilized for data analysis. The study found that the trainin process of the Scratch program had no significant effect on the Computational Thinking Skills and Computing Attitudes of the prospective teachers. However, it was found that the process had a moderate effect only on the CK and TPACK sub-dimensions of prospective teachers TPACK development.

Keywords: Scratch, Technological Pedagogical Content Knowledge, Computational Thinking Skills, Computing Attitude, Distance Education

&

Öz: Bu çalışmanın amacı; Scratch programının ilköğretim matematik öğretmen adaylarının Teknolojik Pedagojik Alan Bilgileri, Bilgi İşlemsel Düşünme Becerileri ve Bilgisayar Bilimi Tutumları üzerindeki etkisini incelemektir. Çalışma 73 ilköğretim matematik öğretmen adayı ile yürütülmüştür. Çalışmada deneme modellerinden tek grup öntest-sontest zayıf deneysel desen tercih edilmiştir. Öğretmen adaylarına Scratch eğitimi verilmeden önce ve verildikten sonra Teknolojik Pedagojik Alan Bilgisi (TPAB) Ölçeği, Bilgi İşlemsel Düşünme Becerileri Ölçeği ve Bilgisayar Bilimi Tutum Ölçeği uygulanmıştır. Verilerin analizi için Wilcoxon işaretler sıralı testi ve bağımlı gruplar t - testi kullanılmıştır. Çalışmanın sonucunda, Scratch programı eğitim sürecinin öğretmen adaylarının Bilgi İşlemsel Düşünme Becerileri üzerinde ve Bilgisayar Bilimi Tutumlarında bir etkisinin olmadığı tespit edilmiştir. Sürecin öğretmen adaylarının Teknolojik Pedagojik Alan Bilgisi (TPAB) gelişimleri üzerine etkisi ise sadece AB ve TPAB alt boyutlarına orta düzeyde etkisinin olduğu tespit edilmiştir.

Anahtar Kelimeler: Scratch, Teknolojik Pedagojik Alan Bilgisi, Bilgi Işlemsel Düşünme Becerileri, Bilgisayar Bilimi Tutumu, Uzaktan Eğitim

Atıf/Cite as: Turan, S. B. & Erdoğan, A. (2025). Scratch's impact on technological pedagogical content knowledge, computational thinking skills, and computing attitudes of prospective elementary mathematics teachers. *Bolu Abant İzzet Baysal Üniversitesi Eğitim Fakültesi Dergisi*, 25(1), 425-445, *https://dx.doi.org/ 10.17240/aibuefd.2025..-1443206.*

İntihal-Plagiarism/Etik-Ethic: Bu makale, en az iki hakem tarafından incelenmiş ve intihal içermediği, araştırma ve yayın etiğine uyulduğu teyit edilmiştir. / This article has been reviewed by at least two referees and it has been confirmed that it is plagiarism-free and complies with research and publication ethics. https://dergipark.org.tr/tr/pub/aibuelt

* Bu çalışma 1. Yazarın doktora tezinden üretilmiştir.

[1] Sorumlu Yazar: Öğr. Gör. Dr. Selva Büşra TURAN, Necmettin Erbakan Üniversitesi, Seydişehir Meslek Yüksekokulu, sbturan@erbakan.edu.tr, ORCID: 0000-0001-8110-022X

[2] Prof. Dr. Ahmet ERDOĞAN, Necmettin Erbakan Üniversitesi, Ahmet Keleşoğlu Eğitim Fakültesi, Matematik Eğitimi ABD, aerdogan@erbakan.edu.tr, ORCID: 0000-0003-2024-4515

1. INTRODUCTION

The world is undergoing rapid change in many aspects, but the most notable change is occurring in the field of technology. Evolving technology has comprehensive effects on education, as it does in civil life, the workplace, and at home (Allen & van der Velden, 2012). Especially with the COVID-19 pandemic period we have experienced in recent years, it has become impossible to think of education and technology as separate concepts (Mumcu, 2022). The most crucial function of education is to cultivate human resources equipped with the skills pertinent to the current era (Doğan, 2014). This role of education also encompasses responding to the educational needs that emerge from the rapid changes and transformations in science and technology (MEB, 2018). Overcoming the rapid changes of today and understanding and making sense of the universe is not only a matter of knowing the areas where technology is used; it also necessitates acquiring and developing skills that aid in adapting to these changes (Erdoğan & Şimşek, 2018). In this context, parallel to the swift developments in technology, educators have made significant progress in making teaching more engaging and easier by successfully integrating emerging technology into traditional teaching methods (Baki, 2018). This process is associated with programming. Programming is the process of appropriately designing and correctly writing sequences of words that enable communication between humans and machines (Erümit &Berigel, 2018). In today's world, individuals are expected to possess 21st-century skills such as problem-solving, critical thinking, analytical thinking, and creativity. One of the most effective methods for developing these skills is through programming. (Yıldız et al., 2017).

Programming is the process of solving problems using a programming language (Moraiti et al., 2022). Programming is not only fundamental in computer science but also a cornerstone of today's and tomorrow's education from which all children can benefit at an early age (Kafai & Burke, as cited in Förster et al., 2018). Programming possesses numerous functions. By learning programming, individuals can more easily adapt to skills such as systematic thinking, problem-solving, discerning relationships between events, and creative thinking. Due to its critical characteristics, it becomes important to instill these skills in children from a young age (Gültepe, 2018). Programming supports and facilitates the development of applications that enhance digital technology and mathematical thinking (Calder, 2018). Appropriate programming languages combined with effective teaching provide alternative support to mathematics education (Feurzeig et al., 1970) and offer opportunities to enhance mathematical thinking through the proper use of digital technologies (Mason, 2005). While the importance of programming in education is evident, the role of teachers and prospective teachers as central figures in this process should not be overlooked. In this regard, the significance and necessity of training prospective teachers, who are equipped with the skills required in the information age and who play critical roles in the educational process, are undeniable (Güleryüz et al., 2020).

When discussing programming, another concept that is inseparably linked and mutually influential is computational thinking (Üzümcü & Bay, 2018). Recognized among the 21st-century skills, computational thinking (Yükseltürk & Üçgül, 2018) is, in its simplest form, defined as using the fundamental concepts of computer science for problem-solving, designing systems, and understanding human behavior (Wing, 2006, p.33). In today's context, where information technology pervades every field and continually expands its boundaries, computational thinking is not just a fundamental skill for computer scientists but is essential for the wider population (Wing, 2006, 2008). From this perspective, it is imperative that prospective teachers are not excluded from developing computational thinking skills. Thus, one of the essential skills that must be imparted to prospective teachers is computational thinking (Aydoğdu, 2020), and it is indicated that computational thinking skills can be developed through programming (Wong &

Cheung, 2020). Additionally, it is noted that programming activities also impact the TPACK development of prospective teachers (Kim & Lee, 2018). Technological Pedagogical Content Knowledge (TPACK) is a framework for thinking about the knowledge teachers need to integrate digital technologies as learning tools in their instructional decision-making (Niess, 2011). The literature also indicates that programming education affects attitudes toward computer technology (Erümit, 2020).

Recently, open-source platforms such as Scratch, Google Blockly, and Code.org, which are user-friendly and supported by numerous visual features, have been developed to facilitate programming and guide and encourage individuals new to learning programming (Aytekin et al., 2018). The fact that users do not need to compile their code or verify the syntax makes these block-based programming tools easier to learn (Holl, 2016). The development of these user-friendly programming tools also provides educators with opportunities to use these programs for teaching mathematics (Germia & Panorkou, 2020).

The Scratch programming tool, in addition to being more user-friendly than older programming languages in terms of adaptability to educational curricula (Shin & Park, 2014), offers future teachers the ease of use in teaching applications as an accessible programming tool (Gabriele et al., 2019). Furthermore, Scratch provides educators with the opportunity to design activities, presentations, games, and animations for use in various fields. The versatility of Scratch for use in subjects beyond coding attracts students' interest and integrates into subjects that are less appealing or challenging, thereby making these subjects more engaging (Aytekin et al., 2018).

In our country, there has been a noticeable increase in recent years in studies conducted using Scratch, a block-based programming tool. In studies carried out with prospective teachers, it is often observed that candidates specializing in computer and instructional technologies are predominantly chosen as the study group. These studies have examined prospective teachers ' attitudes and perceptions towards programming (Arslan & Akçelik, 2019), their motivation and success (Erol, 2015), and their experiences and opinions (Genç & Karakuş, 2011). In studies conducted with prospective mathematics teachers, it has been noted that they have used the Scratch program to create educational games, technology-based mathematical stories, etc., and the developed materials have been analyzed. The research also explores how the process contributes to their professional development, and their awareness of digital mathematics games (Avcu, 2023; Günbaş, 2020; Yıldız Durak & Karaoğlan Yılmaz, 2019). However, it can be said that studies on the use of Scratch in mathematics education are not sufficient (Büyükkarcı, 2019). The limited number of studies conducted with prospective mathematics teachers using the Scratch program and the limited impact of Scratch training on these prospective mathematics teachers are among the reasons for choosing this study. Additionally, the fact that the study coincided with the COVID-19 pandemic allows for an evaluation of the study in the context of distance education.

Purpose of the research

The purpose of this research is to examine the impact of Scratch program training on the Technological Pedagogical Content Knowledge, Computational Thinking Skills, and Computing Attitudes of prospective elementary mathematics teachers. In this context, the impact of the Scratch program education process on prospective teachers' Technological Pedagogical Content Knowledge (TPACK) and its seven sub-dimensions (technological knowledge, content knowledge, pedagogical knowledge, pedagogical content knowledge, technological content knowledge, technological pedagogical knowledge, and technological pedagogical content knowledge) will be evaluated. Additionally, its impact on computational thinking skills and its five sub-dimensions (creativity, collaboration, critical thinking, algorithmic thinking, and problem-solving) and attitudes towards computing attitudes will also be assessed.

In line with this objective, the sub-problems of the research are as follows:
1.Does Scratch have an effect on the Technological Pedagogical Content Knowledge of prospective elementary mathematics teachers?
2.Does Scratch have an effect on the Computational Thinking Skills of prospective elementary mathematics teachers?
3.Does Scratch have an effect on the Computing Attitudes of prospective elementary mathematics teachers?

2. METHOD

2.1. Research design

In this research, a single group pretest-posttest design, a methodology within the experimental research framework, was utilized. Creswell (2003) describes this design as administering a pretest measurement to a group, followed by the experimental treatment, and concluding with a posttest measurement. This design facilitates the assessment of the experimental intervention's impact on a singular cohort. It involves the acquisition of data pertaining to the dependent variable from the same cohort, employing identical evaluative instruments, both prior to and subsequent to the intervention (Büyüköztürk et al., 2020).

2.2. Study group

The study group of this research comprised 73 prospective teachers enrolled in the Algorithm and Programming course during the fourth semester (second year) of the Primary Education Mathematics Teaching Undergraduate Program at a state university. Initially, the prospective teachers were informed about the study. Subsequently, out of the 75 prospective teachers registered for the course, 2 who were absent were excluded from the research. The study was then conducted and completed with the remaining 73 prospective teachers who volunteered to participate.

2.3. Data collection tools

In this study, prospective elementary mathematics teachers were administered the Technological Pedagogical Content Knowledge (TPACK) Scale, Computational Thinking Skills Scale, and Computing Attitude Scale before and after the Scratch program training provided within the scope of the Algorithm and Programming course. The study aimed to evaluate the impact of the training delivered within the course on the prospective teachers.

2.3.1. Technological Pedagogical Content Knowledge (TPACK) Scale: Developed by Çetin and Erdoğan (2018), this scale is designed to measure the TPACK competencies of prospective elementary mathematics teachers. It is a 5-point Likert-type scale consisting of 79 items. The scale comprises seven dimensions: Technological Knowledge (TK), Content Knowledge (CK), Pedagogical Knowledge (PK), Pedagogical Content Knowledge (PCK), Technological Content Knowledge (TCK), Technological Pedagogical Knowledge (TPK), and Technological Pedagogical Content Knowledge (TPACK). The number of items in each dimension and the internal consistency coefficients calculated by the researchers are as follows: TK dimension contains 16 items with an internal consistency coefficient of ,91; CK dimension has 5 items with a coefficient of ,83; PK dimension includes 8 items with a coefficient of ,91; PCK dimension comprises 19 items with a coefficient of ,97; TCK dimension has 8 items with a coefficient of .85; TPK dimension includes 10 items with a coefficient of ,91; and TPACK dimension contains 13 items with a coefficient of ,94. The overall internal consistency coefficient for the scale is ,98. In this study, the Cronbach's alpha reliability coefficient for the TPACK scale was found to be 0,97 for the pretest and 0,98 for the posttest.

2.3.2. Computational Thinking Skills Scale: Developed by Korkmaz et al. (2017), this scale is designed to measure the computational thinking skills of university students. It is a 5-point Likert-type scale ranging from "most negative" to "most positive," consisting of 29 items. The scale encompasses five factors: creativity, collaboration, critical thinking, algorithmic thinking, and problem-solving. The creativity factor includes 8 items with an internal consistency coefficient of 0,843. The collaboration factor comprises 4 items with a coefficient of 0,865. The critical thinking factor contains 5 items with a coefficient of 0,784. The algorithmic thinking factor consists of 6 items with a coefficient of 0,869. The problem-solving factor includes 6 items with a coefficient of 0,727. The overall internal consistency coefficient for the scale is 0,822 (Korkmaz et al., 2015). In this study, the Cronbach's alpha reliability coefficient for the Computational Thinking Skills Scale was found to be 0.91 for the pretest and 0.93 for the posttest.

2.3.3. Computing Attitude Scale: The Computing Attitude Scale, developed by Yadav et al. (2014), is a measure consisting of 21 Likert-type items across five sub-dimensions. The scale includes levels of agreement ranging from 'strongly agree' to 'strongly disagree'. The reliability of the scale was found to be $\alpha = 0.76$. The scale was adapted into Turkish by Turan (2022) after obtaining the necessary permissions from the original researchers. During the Turkish adaptation of the scale, both exploratory and confirmatory factor analyses were conducted, confirming the scale's unidimensional structure with 19 items. The reliability of the scale was assessed using the Cronbach's Alpha coefficient. The 19-item scale achieved a high reliability with a Cronbach's Alpha coefficient of 0,890. In this study, the Cronbach's alpha reliability coefficient for the Computing Attitude Scale was found to be 0,88 for the pretest and 0,89 for the posttest.

2.4. Procedure

The research was conducted as part of the "Algorithm and Programming" course taken by prospective elementary mathematics teachers during the spring semester. The course, scheduled for two hours per week, was taught to prospective elementary mathematics teachers in two separate sections using the same content and methodology. The semester spanned 15 weeks, but due to a week-long break for Ramadan, the course activities were carried out over 14 weeks. The study was conducted synchronously via the university's distance education system due to the COVID-19 pandemic. This synchronous educational approach facilitated an interactive process with the prospective teachers.

In the first week, scales were prepared on Google Forms and sent to the prospective elementary mathematics teachers for the pretest application. Each of the distributed scales was completed by the prospective teachers. In the second and third weeks, examples of algorithm design and flowcharts were presented. In the fourth week, information about computational thinking concepts was provided. Starting from the fifth week, instruction on the Scratch programming began. In the week when the introduction to the Scratch 3.0 program commenced, in addition to the live lesson presentations, the researcher conducted video recordings outside of class hours, showcasing the features of the Scratch program. These instructional videos, recorded by the researcher, were uploaded to the researcher's personal YouTube channel, and the links to these videos were shared with the prospective elementary mathematics teachers via the distance education system. Following the distribution of the video links, the prospective elementary mathematics teachers were informed through both the distance education system and a WhatsApp group, and they were encouraged to watch the video content before the next live class session. Any queries about the content of the videos were addressed during live class sessions, where the prospective elementary mathematics teachers could ask the researcher questions to clarify their understanding of the Scratch 3.0 program.

Starting from the eighth week, the prospective teachers began preparing Scratch projects appropriate for the learning outcomes specified in the 2018 Mathematics Teaching Program (Middle School Grades 5, 6, 7, and 8), starting from the 5th-grade level and subsequently progressing to the 6th, 7th, and 8th grades. Prospective teachers were allowed to develop their Scratch projects in various genres such as stories, animations, or games, without any restrictions imposed. The Scratch projects, created by the prospective teachers based on different learning outcomes, were uploaded to designated studios, enabling them to view, evaluate, and comment on projects prepared with various learning outcomes and develop ideas for projects for the next grade level.

From the 13th week onwards, lesson plans integrated with the Scratch program, aligned with the learning outcomes in the 2018 Mathematics Teaching Program (Middle School Grades 5, 6, 7, and 8), were presented to the prospective teachers. In the preparation of the plan presented to prospective teachers, the Ministry of National Education's Directive on the Planned Execution of Educational and Instructional Activities (MEB, 2003) and relevant literature were utilized. When preparing the lesson plan, the following formal elements were included: grade level, learning area, learning outcomes, possible duration, and the methods-techniques to be used. In the introduction section, attention and motivation suitable for the students' readiness level, assessment of students' prerequisite knowledge, presentation of the learning outcomes, and transition to the application part were included. The development section was designed to incorporate cues, feedback correction, reinforcement, and reasoning. The assessment and evaluation sections were designed to answer the question "What have we learned?" (MEB, 2003; Baykul, 1998; Sönmez, 2012).

At the end of the 14-week course period, the prospective teachers were asked to prepare and upload to the distance education system, within two weeks, lesson plans integrated with the Scratch program, suitable for the grade level and learning outcomes they had chosen from the 2018 Mathematics Teaching Program. At the end of the course, the scales initially administered as a pretest were resent to the prospective teachers for the posttest, and each of the distributed scales was again completed by the prospective teachers.

2.5. Data analysis

A single-group pretest-posttest experimental design was used from the trial models. In the research, the normal distribution of the scores obtained by prospective teachers from the Computing Attitude Scale, Technological Pedagogical Content Knowledge (TPACK) Scale, and Computational Thinking Skills Scale was examined to test their compliance with parametric test assumptions. Given that the sample size exceeded 50, the Kolmogorov-Smirnov test was utilized to evaluate the suitability of the data for a normal distribution (Büyüköztürk, 2012).

The results of the Kolmogorov-Smirnov test for the scales administered are presented in Table1.

Table 1.

Prospective Teachers Pretest and Posttest Scores From Scales and the Results of Normality Tests.

	Kolmogorov – Smirnov		
	Statistics	**sd**	**p**
Computing Attitude Scale Pretest-Posttest	,122	73	,009
TPACK Scale Pretest-Posttest	,079	73	,20
Computational Thinking Skills Pretest-Posttest	,067	73	,20

*p<,05

Upon examining Table 1, it was observed that the significance level for the differences in pretest and posttest scores of the Computing Attitude Scale was below ,05 (p = ,009), indicating a non-normal distribution of the data. Consequently, the use of non-parametric statistical techniques was deemed appropriate. The Wilcoxon signed-rank test, a nonparametric test, was utilized to determine the difference in mean scores obtained from the scale. Conversely, for the TPACK Scale and Computational Thinking Skills Scale, the significance level was above ,05 (p = ,20), suggesting a normal distribution of the data. Therefore, the application of parametric statistical techniques was decided upon. To ascertain the difference in mean scores derived from these scales, the dependent groups t-test, a parametric test, was utilized.

Effect size is another statistical consideration in interpreting hypothesis test results when comparing mean scores. One of the statistics for effect size is the eta-squared (η^2) correlation coefficient. Eta-squared, which does not require the assumption of linearity between variables, indicates the extent to which the independent variable influences the dependent variable. It is also referred to as effect size, demonstrating the proportion of the total variance in the dependent variable explained by the independent variable or factor (Büyüköztürk, 2012).
In the conducted research, to calculate the effect size in the knowledge types where a significant difference was observed between pretest and posttest scores, the eta-squared (η^2) coefficient was computed. The eta-squared coefficient ranges between 0 and 1 and is interpreted as a "small" effect at ,01, a "medium" effect at ,06, and a "large" effect at ,14. The eta-squared effect size statistic for the related t-test is calculated using the following formula (Büyüköztürk, 2012; Pallant, 2001).
The formula for calculating the eta-squared (η^2) for the dependent groups t-test is:

$$\eta^2 = \frac{t^2}{t^2 + N - 1}$$

3. FINDINGS

Findings on the impact of Scratch on prospective elementary mathematics teachers "Technological Pedagogical Content Knowledge"

The responses of 73 prospective teachers to the Technological Pedagogical Content Knowledge Scale were analyzed at the beginning of the research process during the pretest and at the end during the posttest. To determine the difference between the mean scores obtained from the scale, the dependent groups t-test, a parametric test, was utilized. The t-test results, examining whether there was a significant difference in the Technological Pedagogical Content Knowledge of the prospective teachers at the beginning and end of the study, are presented in Table 2.

Table 2.
Results of the T-Test for Dependent Groups Regarding Pre-Test and Post-Test Scores of TPACK

Measurement	N	$\overline{X}$	S	sd	t	p
Pretest	73	319,16	37,20	72	-1,38	,17
Posttest	73	328,50	37,87			

*p< ,05

Upon examining Table 2, it is observed that there is no significant difference between the pretest and posttest mean scores of prospective teachers on the Technological Pedagogical Content Knowledge (TPACK) scale ($t_{(72)} = -1,38$, $p > ,05$). The results of the dependent groups t-test for the sub-dimensions of the TPACK scale are presented in Table 3.

Table 3.
Results of the T-Test For Dependent Groups Related to Pretest and Posttest Scores of the Sub-Dimensions of the TPACK

Dimensions	Measurement	N	$\overline{X}$	S	sd	t	p	η^2
TK	Pretest	73	63,90	8,59	72	-,85	0,395	-
	Posttest	73	65,21	8,90				
CK	Pretest	73	18,75	3,57	72	-2,37	,020*	0,084
	Posttest	73	20,19	3,15				
PK	Pretest	73	33,54	4,48	72	-,55	,582	-
	Posttest	73	33,97	4,13				
PCK	Pretest	73	81,69	8,97	72	-,63	,526	-
	Posttest	73	82,76	9,61				
TCK	Pretest	73	32,28	4,19	72	-1,15	,252	-
	Posttest	73	33,13	4,18				
TPK	Pretest	73	40,53	6,02	72	-1,24	,218	-
	Posttest	73	41,75	5,37				
TPACK	Pretest	73	48,43	8,51	72	-2,18	,032*	0,062
	Posttest	73	51,46	7,78				

*p< ,05

Upon reviewing Table 3, the results of the dependent groups t-test for the differences in pretest-posttest mean scores indicate that there is a significant difference in the Content Knowledge (CK) sub-dimension ($t_{(72)}$ = -2,73, p < ,05) and the Technological Pedagogical Content Knowledge (TPACK) sub-dimension ($t_{(72)}$ = -2,18, p < ,05) of the TPACK scale. However, no significant differences were observed in the pretest-posttest mean scores for the Technological Knowledge (TK) sub-dimension ($t_{(72)}$ = -,85, p > ,05), Pedagogical Knowledge (PK) sub-dimension ($t_{(72)}$ = -,55, p >,05), Pedagogical Content Knowledge (PCK) sub-dimension ($t_{(72)}$ = -,63, p > ,05), Technological Content Knowledge (TCK) sub-dimension ($t_{(72)}$ = -1,15, p > ,05), and Technological Pedagogical Knowledge (TPK) sub-dimension ($t_{(72)}$ = -1,24, p > ,05). This finding suggests that learning Scratch, a block-based programming tool, positively impacts the CK and TPACK sub-dimensions of prospective teachers.

For the dependent groups t-test, the eta-squared (η^2) values were calculated as CK > 0,084 and TPACK > 0,062. These results indicate that the integration process of Scratch had a moderate effect on the only CK and TPACK sub-dimensions of the prospective teachers.

Findings on the impact of Scratch on prospective elementary mathematics teachers "Computational Thinking Skills"

The responses of 73 prospective teachers to the Computational Thinking Skills Scale were analyzed at the beginning of the research process during the pretest and at the end during the posttest. To determine the difference between the mean scores obtained from the scale, the dependent groups t-test, a parametric test, was utilized. The t-test results, examining whether there was a significant difference in the computational thinking skills of the prospective teachers at the beginning and end of the study, are presented in Table 4.

Table 4.
Results of the T-Test for Dependent Groups Related to Pretest and Posttest Scores of Computational Thinking Skills

Measurement	N	$\overline{X}$	S	sd	t	p
Pretest	73	115,86	12,65			
Posttest	73	117,43	14,40	72	-,75	,451

*p<,05

Upon reviewing Table 4, it is observed that there is no significant difference between the pretest and posttest mean scores of prospective teachers on the Computational Thinking Skills Scale ($t_{(72)}$ = -,75, p > ,05). The results of the dependent groups t-test for the sub-dimensions of the Computational Thinking Skills Scale are presented in Table 5.

Table 5.
Results of the T-Test for Dependent Groups Related to Pretest and Posttest Scores of the Sub-Dimensions of the Computational Thinking Skills Scale

Dimensions	Measurement	N	$\overline{X}$	s	sd	t	p
Creativity	Pretest	73	33,32	3,37	72	-,87	,38
	Posttest	73	33,84	3,99			
Algorithmic Thinking	Pretest	73	23,53	3,35	72	-,97	,33
	Posttest	73	24,06	3,80			
Collaboration	Pretest	73	15,93	3,22	72	,47	,63
	Posttest	73	15,68	3,27			
Critical Thinking	Pretest	73	18,71	3,17	72	-1,57	,12
	Posttest	73	19,47	3,14			
Problem Solving	Pretest	73	24,35	3,88	72	,00	,99
	Posttest	73	24,35	4,05			

*p<,05

According to the results of the dependent groups t-test for the difference in pretest-posttest mean scores, there is no significant increase in any of the sub-dimensions of the Computational Thinking Skills Scale for prospective teachers at the end of the research process.

Upon examining Table 5, it is observed that there is no significant difference in the pretest-posttest mean scores for the sub-dimensions of the Computational Thinking Skills Scale: creativity ($t_{(72)}$ = -,87, p > ,05), algorithmic thinking ($t_{(72)}$ = -,97, p > ,05), collaboration ($t_{(72)}$ = ,47, p > ,05), critical thinking ($t_{(72)}$ = -1,57, p > ,05), and problem-solving ($t_{(72)}$ = ,00, p > ,05). This finding suggests that learning Scratch, a block-based programming tool, does not have an impact on the computational thinking skills of prospective teachers.

Findings on the impact of Scratch on prospective elementary mathematics teachers "Computing Attitudes"

The responses of 73 prospective teachers to the Computing Attitude Scale were analyzed at the beginning of the research process during the pretest and at the end during the posttest. To determine the difference between the mean scores obtained from the scale, the Wilcoxon signed-rank test, a nonparametric test, was utilized. The Wilcoxon signed-rank test results, examining whether there was a significant difference in the attitudes towards computing of the prospective teachers at the beginning and end of the study, are presented in Table 6.

Table 6.

Results of the Wilcoxon Signed-Rank Test on the Computing Attitude Scale Scores of Prospective Teachers at the Beginning and End of the Study

Posttest-Pretest Measurement	N	Mean rank	Sum of ranks	z	p
Negative Rank	36	27,64	995,00	1,08	,27
Positive Rank	32	42,22	1351,00		
Ties	5				

Upon examining Table 6, it is observed that there is no significant difference between the pretest and posttest scores of the prospective teachers on the Computing Attitude Scale (z = 1,08; p > ,05). This finding suggests that learning Scratch, a block-based programming tool, does not have an impact on the attitudes of prospective teachers towards computing.

4. DISCUSSION and CONCLUSION

The effect of Scratch on the development of prospective teachers Technological Pedagogical Content Knowledge (TPACK) in relation to the first sub-problem of the research was assessed using the TPACK scale, administered at the beginning (pretest) and end (posttest) of the educational process. When comparing the scores obtained from the overall scale and its sub-dimensions, no significant difference was found in the overall scores; however, significant differences emerged in only the Content Knowledge (CK) and Technological Pedagogical Content Knowledge (TPACK) sub-dimensions. To determine which of these two sub-dimensions exhibited the greatest effect, the eta-squared (η^2) values for both were examined; it was observed that both CK (0.084) and TPACK (0.062) sub-dimensions had a moderate effect size.

Content Knowledge is defined as the knowledge about the subject matter that is to be learned or taught (Mishra & Koehler, 2006). Therefore, it is anticipated that prospective teachers must possess sufficient content knowledge related to the selected topic or learning outcome to effectively develop Scratch projects aligned with these chosen objectives. In this context, it can be asserted that the process of conducting research on the content knowledge related to the selected topic or learning outcome, identifying and addressing any gaps, and integrating mathematical concepts and information with the Scratch program while developing Scratch projects and preparing lesson plans, contributes to the development of Content Knowledge.

Pedagogical Knowledge pertains to the processes and practices of teaching and learning, encompassing knowledge about student learning, classroom management, lesson plan development and implementation, and student assessment (Mishra & Koehler, 2006). A teacher equipped with this knowledge understands how students learn, how to measure and evaluate learning outcomes, and effective classroom communication techniques (Çetin & Erdoğan, 2018). The lack of a significant difference in the Pedagogical Knowledge dimension is thought to be due to the prospective teachers being second-year students, which implies limited pedagogical knowledge. Additionally, the inability to implement the projects and lesson plans developed post-training during the research process in a classroom setting, due to the COVID-19 pandemic, and the lack of practical experience in this regard, are considered contributing factors to the absence of significant improvement in Pedagogical Knowledge.

Pedagogical Content Knowledge involves understanding which teaching approaches are suitable for specific content and how elements of content can be organized for more effective instruction (Mishra & Koehler, 2006). In the Undergraduate Program for Primary Mathematics Teaching, courses directly related to pedagogical content knowledge, such as Teaching of Numbers, Geometry and Measurement Instruction, Algebra Teaching, Probability and Statistics Teaching, and Assessment and Evaluation in Education, are covered at the third-year level (V and VI semesters) (YÖK, 2018). Therefore, the absence of a significant difference in the Pedagogical Content Knowledge sub-dimension of the scale can be attributed to the fact that the prospective teachers are still in their second year (IV semester) and have not yet taken courses related to the instruction of their subject matter, nor have they experienced classroom teaching. This lack of experience is considered a contributing factor to the observed result.

It has been observed that there is no significant difference in the Technological Knowledge (TK), Technological Content Knowledge (TCK), and Technological Pedagogical Knowledge (TPK) sub-dimensions of the TPACK scale. Technological Knowledge refers to the knowledge and skills teachers need to have in using technology (Çetin & Erdoğan, 2018). In addition, the fact that there was no significant difference in these three sub-dimensions may be due to the fact that TB is strongly related to TCK and TPK and thus affects their development (Kim & Lee, 2018). It can be assumed that pre-service teachers have taken courses such as Information Technologies, Instructional Technologies, Computer Assisted Mathematics Teaching before taking the Programming and Algorithm course in which they are included within the scope of the research, so they have prior knowledge about using technology. In the literature, block-based visual programming tools are noted for providing an easy-to-use editing interface to promote programming learning for novice students of various ages (Holl, 2016; Hu et al., 2021). These reasons are thought to contribute to the lack of significant difference in the Technological Knowledge sub-dimension. It is stated that prospective teachers often struggle with learning programming initially, and this difficulty can hinder their development of technological knowledge (Kim & Lee, 2018). It is likely that the prospective teachers in this study also lack experience in programming, which may affect their technological knowledge development. Conducting pilot applications not included in this study could provide prospective teachers with extended experience using programming tools, potentially reducing the difficulties they face in learning programming. Consequently, this could support their development of technological knowledge.

Technological Pedagogical Knowledge involves understanding how teaching will change in the expected direction as a result of using specific technologies (Mishra & Koehler, 2006). Considering that the prospective teachers learned and applied the subject within a relatively short period of one semester, it is believed that the lack of experience contributed to the absence of significant difference in the Technological Pedagogical Knowledge sub-dimension. Similarly, Tatlı et al. (2016) in their study attribute the lower scores in the Technological Pedagogical Knowledge sub-dimension compared to other dimensions to a lack of experience. Technological Content Knowledge (TCK), which refers to the knowledge about the relationship between technology and course content (Mishra & Koehler, 2006), also showed no significant difference in this sub-dimension. Kim and Lee (2018) state that the problems arising from using programming in subject matter and curriculum during the process of designing programming-based courses can hinder the development of technological content knowledge (TCK). To overcome these issues, it is necessary to analyze TPACK examples based on programming. In this context, the process of integrating Scratch into lesson plans, which started in the 13th week and lasted for 3 weeks, may not have been sufficient to develop the technological content knowledge of prospective teachers. It is considered beneficial to conduct a detailed examination of the curriculum and discuss how programming (Scratch) can be integrated into the subject matter and curriculum with prospective teachers over a longer period, providing examples.

However, a significant difference was observed in the Technological Pedagogical Content Knowledge (TPACK) sub-dimension of the scale. TPACK is a form of knowledge that transcends the three components of content, pedagogy, and technology (Mishra & Koehler, 2006). Günbaş (2020) stated that prospective teachers are predominantly trained using traditional methods throughout their education and have limited opportunities to use educational technology tools. In this context, in a study conducted with prospective mathematics teachers using the Scratch program, it was concluded that the process aided in the development of their TPACK. As indicated in the literature, Yıldız Durak and Karaoğlan Yılmaz (2019) investigated the educational digital game designs in the Scratch program and the perspectives of prospective mathematics teachers towards mathematics teaching. Their study concluded that the process of designing digital games enhanced the prospective teachers' technological proficiency, pedagogical development, and content knowledge. This outcome aligns with the findings of Günbaş (2020). Therefore, it can be stated that the use of Scratch, given appropriate educational opportunities, has the potential to enhance the technological pedagogical content knowledge of prospective mathematics teachers.

The impact of Scratch on prospective teachers Computational Thinking Skills, in relation to the second sub-problem of the research, was assessed using the "Computational Thinking Skills Scale" administered at the beginning (pretest) and end (posttest) of the educational process. When comparing the scores obtained from the overall scale and its sub-dimensions, no significant difference was found between the scores. In the literature, a similar study conducted by Aydoğdu (2020) titled "The Effect of Block-Based Programming Activities on Pre-Service Teachers' Self-Efficacy Perceptions and Computational Thinking Skills" found that block-based programming activities did not have an impact on the computational thinking skills of prospective teachers. The study suggested that this result might be due to the prospective teachers ' average scores being above the midpoint of the scale even before the intervention. Similarly, in this research, it was observed that the initial average scores of the prospective teachers on the scale were above the midpoint of the computational thinking skills scale. As evaluated in Aydoğdu's (2020) study, the pre-intervention average scores of the prospective teachers being above the midpoint of the scale is considered to be the reason why no significant difference was found between the scores obtained from the overall scale and its sub-dimensions.

Contrary to the findings of this research, there are studies indicating that block-based programming enhances computational thinking skills (Ceylan, 2020; Kim et al., 2012; Papadakis & Kalogiannakis, 2019; Rodriguez-Martinez et al., 2020; Yünkül et al., 2019). Additionally, the literature includes research suggesting that programming instruction is among the approaches used to develop computational thinking skills (Alsancak Sarıkaya, 2019), and that programming skills contribute to the development of critical thinking, algorithmic thinking, problem-solving, and creativity (Akçay & Çoklar, 2016).

The impact of Scratch on prospective teachers' attitudes towards computing, in relation to the third sub-problem of the research, was assessed using the "Computing Attitude Scale" administered at the beginning (pretest) and end (posttest) of the educational process. Upon comparing the scores obtained from the scale, it was concluded that there was no significant difference in the attitudes towards computing. Köseoğlu et al. (2007), in their study conducted over 10 weeks with biology prospective teachers, concluded that a computer course did not positively affect the candidates' attitudes towards computers. They attributed this outcome to the course duration being insufficient for developing attitudes. Similarly, in this research, it can be considered that the duration of the training and practices conducted over a semester may not have been sufficient to develop positive attitudes towards computer science among prospective teachers. Akçay and Çoklar (2016) suggest that programming skills, being production-centered, can yield positive outcomes through the motivation derived from production. Another possible reason for the lack of significant change in prospective teachers ' attitudes towards computer science could be that the process was not sufficient to develop programming skills.

While the research didn't find a positive impact of Scratch, a block-based programming tool, on attitudes towards computer science, other studies have shown positive effects of physical programming. For instance, Theodoropoulos et al. (2018) reported that students' positive attitudes towards computer science improved after a 10-week study involving Arduino robotics activities. Coşar

(2013) also highlighted that individuals with strong programming skills tend to have a more positive attitude towards computer science (cited in Akçay & Çoklar, 2016). Furthermore, Khoja et al. (2012), in a 4-week educational camp with 30 female students in grades 7 and 8, observed an increase in interest and attitudes towards computer science among students after receiving training in various topics, including computer science introduction, Lego Mindstorms, Alice programming, and hardware. Erümit (2020) conducted a study with sixth grade students using Scratch and concluded that the process supported the development of positive attitudes towards computer technology. In this study conducted with pre-service teachers, it can be thought that the reason why there is no effect on the attitude towards computing may also be due to the age factor.

As a result, in today's era where traditional teaching methods are being abandoned, the impact of changes and transformations in instructional technologies on various emerging learning and teaching approaches, and their effects on educational processes, is inevitable (Mumcu, 2022). Teachers are seen as the most critical stakeholders in the utilization of many innovations in the field of education within learning and teaching processes (Admiraal et al., 2017). The success of potential changes in the education system is directly related to teachers being open to innovations and adapting to gain necessary skills (Fullan, 1991). Moreover, the successful use of technology in the learning-teaching process is also related to teachers' acceptance of technology and their intention to use it (Admiraal et al., 2017). In this study conducted with prospective teachers, who will be the future educators, it was observed that the Scratch program training process did not have an impact on the computational thinking skills and attitudes towards computing of the prospective teachers. However, the process was found to have a moderate effect only on the CK and TPACK sub-dimensions of their technological pedagogical content knowledge (TPACK) development.

Moreover, the frequent inclusion of concepts such as algorithms, flowcharts, programming, and coding in the 2024 middle school mathematics curriculum demonstrates efforts to integrate programming into the curriculum. In this context, conducting studies using different programming tools, in addition to the Scratch programming tool used in this study, will be beneficial for prospective teachers who will be the future implementers of the curriculum.

Reference

Admiraal, W., Louws, M., Lockhorst, D., Paas, T., Buynsters, M., Cviko, A., ... & Kester, L. (2017). Teachers in school-based technology innovations: A typology of their beliefs on teaching and technology. Computers & Education, 114, 57-68.

Akçay, A. & Çoklar, A. N. (2016). Bilişsel becerilerin gelişimine yönelik bir öneri: Programlama eğitimi [A proposal for the development of cognitive skills: Programming training]. In A. İşman, H. F. Odabaşı & B. Akkoyunlu (Eds.), Eğitim teknolojileri okumaları 2016 (pp. 121- 140). TOJET-Sakarya University.

Allen, J. P., & van der Velden, R. K. W. (2012). Skills for the 21st century: implications for education. Researchcentrum voor Onderwijs en Arbeidsmarkt, Faculteit der Economische Wetenschappen. ROA Research Memoranda No. 11 https://doi.org/10.26481/umaror.2012011

Alsancak Sırakaya, D. (2019). Programlama öğretiminin bilgi işlemsel düşünme becerisine etkisi. [The effect of programming teaching on computational thinking]. *Turkish Journal of Social Research, 23*(2), 575-590.

Arslan, K., & Akçelik, M. (2019). Programlama eğitiminde Scratch'in kullanılması: öğretmen adaylarının tutum ve algıları [Using scratch in programming language: teacher candidates' attitudes and perceptions]. *Ulusal Eğitim Akademisi Dergisi 3*(1), 41-61. https://doi.org/10.32960/uead.455502

Avcu, S. (2023). Matematik Öğretmen Adaylarının Scratch ile Tasarlanan Dijital Matematik Oyunlarıyla İlgili Farkındalıkları [Prospective mathematics teachers' awareness about digital mathematics games designed with scratch]. Van Yüzüncü Yıl University Journal of Education, *20*(1), 126-149. https://doi.org/10.33711/yyuefd.1178451

Aydoğdu, Ş. (2020). Blok tabanlı programlama etkinliklerinin öğretmen adaylarının programlamaya ilişkin öz yeterlilik algılarına ve hesaplamalı düşünme becerilerine Etkisi. [The effect of block-based programming activities on pre-service teachers' computer programming self-efficacy and computational thinking skills]. *Educational Technology Theory and Practice*, *10* (1), 303-320. https://doi.org/10.17943/etku.649585

Aytekin, A., Sönmez Çakır, F., Yücel, Y.B., & Kulaözü, İ. (2018). Geleceğe yön veren kodlama bilimi ve kodlama öğrenmede kullanılabilecek bazı yöntemler [Coding science directed to future and some methods to be available and coding learned]. *Eurasian Journal of Social and Economic Research*, *5*(5), 24-41.

Baki, A. (2018). *Matematiği öğretme bilgisi [The knowledge of teaching maths].* Pegem Akademi.

Baykul, Y. (1998). *İlköğretim birinci kademede matematik öğretimi*. Milli Eğitim Basımevi.

Büyükkarcı, A. (2019). *Kodlama ile zenginleştirilmiş 5E modelinin 4. sınıf matematik başarısına, kalıcılığına ve tutumuna etkisi* [*The effect of 5E model enriched with coding on 4th grade mathematics achievement, permanence and attitude*] (Doctoral thesis). Burdur Mehmet Akif Ersoy Üniversitesi.

Büyüköztürk, Ş. (2012). *Sosyal bilimler için veri analizi el kitabı [Data analysis for social sciences hand book]*. Pegem Akademi.

Büyüköztürk, Ş., Çakmak, E. K., Akgün, Ö. A., Karadeniz, Ş., & Demirel, F. (2020). *Eğitimde bilimsel araştırma yöntemleri [Scientific research methods in education].* Pegem Akademi.

Calder, N. (2018). Using Scratch to facilitate mathematical thinking'. *Waikato Journal of Education, 23*(2), 43-58. https://doi.org/10.15663/wje.v23i2.654

Ceylan, V. K. (2020). *Senaryo temelli Scratch öğretim programının öğrencilerin bilgi işlemsel düşünme becerilerine, problem çözme ve programlama ünitesi erişilerine etkisi [The effect of scenario based learning scratch curriculum on students' computational thinking skills and learning outcomes of problem solving and programming unit]* (Doctoral thesis). Adnan Menderes Üniversitesi.

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 thesis). Gazi Üniversitesi.

Creswell, J. W. (2003). *Research design gualitative, quantitative and mixed methods approaches.* Sage publications.

Çetin, İ. & Erdoğan, A. (2018). Development, validity and reliability study of technological pedagogical content knowledge (TPACK) efficiency scale for mathematics teacher candidates. *International Journal of Contemporary Educational Research, 5*(1), 50-62.

Doğan, S. (2014). Eğitimin işlevleri. In C.T. Uğurlu (Ed.), *Eğitim bilimine giriş* (pp.39-53). Eğiten Kitap.

Erdoğan, T. ve Şimşek, A. (2018). Programlama öğretimi ve drama. In Y. Gülbahar & H. Karal (Eds.), *Kuramdan uygulamaya programlama öğretimi* (pp. 237-269). Pegem Akademi.

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 achievement]* (Doctoral thesis). Anadolu Üniversitesi.

Erümit, A. K. (2020). Effects of different teaching approaches on programming skills. *Education and Information Technologies*, *25*(2), 1013-1037.

Erümit, A. K. & Berigel, M. (2018). Programlama dillerinin tarihi ve programlama öğretimi. In Y. Gülbahar & H. Karal (Eds.), *Kuramdan uygulamaya programlama öğretimi* (pp. 2-36). Pegem Akademi.

Feurzeig, W., Papert, S., Bloom, M., Grant, R., & Solomon, C. (1970). Programming languages as a conceptual framework for teaching mathematics. *ACM SIGCUE Outlook*, *4*(2), 13-17. https://doi.org/10.1145/965754.965757

Förster, E. C., Förster, K. T., & Löwe, T. (2018, April). Teaching programming skills in primary school mathematics classes: An evaluation using game programming. In *2018 IEEE Global Engineering Education Conference (EDUCON)* (pp. 1504-1513). IEEE.

Fullan, M. G. (1991). The new meaning of educational change. New York: Teachers College Press.

Gabriele, L., Bertacchini, F., Tavernise, A., Vaca-Cárdenas, L., Pantano, P., & Bilotta, E. (2019). Lesson planning by computational thinking skills in Italian pre-service teachers. *Informatics in Education, 18*(1), 69-104.

Genç, Z. & Karakuş, S. (2011, Eylül). *Learning through design: using scratch in instructional computer games design* [Conference presentation]. 5th International Computer & Instructional Technologies Symposium, Elazığ.

Gültepe, A. (2018). Kodlama öğretimi yapan bilişim teknolojileri öğretmenleri gözüyle öğrenciler kodluyor [Make coding teaching by ıct teachers eye "students are encoding"]. *International Journal of LeadershipTraining, 2*(2), 50-60.

Güleryüz, H., Dilber, R. & Erdoğan, İ. (2020). STEM uygulamalarında öğretmen adaylarının kodlama eğitimi hakkındaki görüşleri [Prospective teachers' views on coding training in stem applications]. *Ağrı İbrahim Çeçen Üniversitesi Sosyal Bilimler Enstitüsü Dergisi 6* (1), 71-83. https://doi.org/10.31463/aicusbed.610909

Günbaş, N. (2020). Prospevtive mathematics teachers' technological pedagogical content knowledge improvement via creating technology-based mathematics stories. *International Journal of Psychology and Educational Studies, 7* (3), 209-222.

Holl, A. (2016). Code girl [Mater's theses]. Santa Clara University.

Hu, Y., Chen, C. H., & Su, C. Y. (2021). Exploring the effectiveness and moderators of block-based visual programming on student learning: A meta-analysis. *Journal of Educational Computing Research*, *58*(8), 1467-1493.

Khoja, S., Wainwright, C., Brosing, J., & Barlow, J. (2012). Changing girls' attitudes towards computer science. *Journal of Computing Sciences in Colleges*, *28*(1), 210-216.

Kim, H., Choi, H., Han, J. & So, H.J. (2012) 'Enhancing teachers' ICT capacity for the 21st century learning environment: Three cases of teacher education in Korea', *Australasian Journal of Educational Technology*, 28(6), 965-982. https://doi.org/10.14742/ajet.805

Kim, S. W., & Lee, Y. (2018). Development and application of the TPACK-P education program for pre-service teachers' TPACK. *International Journal of Engineering & Technology*, *7*(3.34), 636-643.

Korkmaz, Ö., Çakır, R., Özden, M. Y., Oluk, A., & Sarıoğlu, S. (2015). Bireylerin bilgisayarca düşünme becerilerinin farklı değişkenler açısından incelenmesi [Investigation of ındividuals' computational thinking skills in terms of different variables]. *Ondokuz Mayıs Üniversitesi Eğitim Fakültesi Dergisi*, *34* (2), 68-87. https://doi.org/10.7822/omuefd.215276

Korkmaz, Ö., Çakır, R., & Özden, M. (2017). A validity and reliability of the computational thinking scales (CTS). *Computers in Human Behavior*,72,1-43. https://doi.org/10.1016/j.chb.2017.01.005

Köseoğlu, P., Yılmaz, M., Gerçek, C., & Soran, H. (2007). Bilgisayar kursunun bilgisayara yönelik başarı, tutum ve öz-yeterlik inançları üzerine etkisi [Effects of computer courses on achievement, attitudes and self-efficacy beliefs regarding computers]. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi 33*, 203-209.

Mason, J. (2005). Mediating mathematical thinking with e-screens. In S. Johnston Wilder ve D. Pimm (Eds.), *Teaching Secondary Mathematics with ICT.* (pp. 81-100). Berkshire, UK: Open University Press.

Milli Eğitim Bakanlığı (MEB). (2003). Millî Eğitim Bakanlığı eğitim ve öğretim çalışmalarının plânlı yürütülmesine ilişkinyönerge. http://mevzuat.meb.gov.tr/dosyalar/284.pdf adresinden 10 Eylül 2021 tarihinde alınmıştır.

Milli Eğitim Bakanlığı (MEB) (2018). Matematik dersi öğretim programı (İlkokul ve Ortaokul 1, 2, 3, 4, 5, 6, 7 ve 8. sınıflar). 10 Eylül 2021, https://mufredat.meb.gov.tr/ProgramDetay.aspx?PID=329

Mishra, P. & Koehler, M. J. (2006). Technological pedagogical content knowledge: a framework for ıntegrating technology in teacher knowledge. *Teachers College Record*, *108*(6), 1017-1054.

Moraiti, I., Fotoglou, A., & Drigas, A. (2022). Coding with Block Programming Languages in Educational Robotics and Mobiles, Improve Problem Solving, Creativity & Critical Thinking Skills. *International Journal of Interactive Mobile Technologies*, *16*(20).

Niess, M. L. (2011). Investigating TPACK: Knowledge growth in teaching with technology. *Journal of Educational Computing Research, 44*(3), pp. 299-317. doi:10.2190/EC.44.3.c

Pallant, J. (2001). SPSS Survival Manual. A Step-by-Step Guide to Data Analyses Using SPSS for Windows. Philadelphia, *PA: Open University Press.*

Papadakis, S., & Kalogiannakis, M. (2019). Evaluating a course for teaching İntroductory programming with Scratch to pre-service kindergarten teachers. *International Journal of Technology Enhanced Learning*, *11*(3), 231-246.

Rodríguez-Martínez, J. A., González-Calero, J. A. & Sáez-López, J. M. (2020). Computational thinking and mathematics using Scratch: an experiment with sixth-grade students, *Interactive Learning Environments 28*(3), 316-327. https://doi.org/10.1080/10494820.2019.1612448

Shin, S. & Park, P (2014). A study on the effect affecting problem solving ability of primary students through the Scartch programming. *Advanced Science and Technology Letters 59*, 117-120.

Sönmez, V. (2012). *Program geliştirmede öğretmen el kitabı*. Anı Yayıncılık.

Tatlı, Z., Akbulut, H. İ., & Altınışık, D. (2016). Öğretmen adaylarının teknolojik Pedagojik alan bilgisi özgüvenlerine web 2.0 araçlarının etkisi [The Impact of Web 2.0 Tools on Pre-Service Teachers' Self Confidence Levels About TPCK]. *Turkish Journal of Computer and Mathematics Education*, *7*(3), 659. https://doi.org/10.16949/turkbilmat.277878

Theodoropoulos, A., Leon, P., Antoniou, A., ve Lepouras, G. (2018, October). Computing in the physical world engages students: Impact on their attitudes and self-efficacy towards Computer Science through robotic activities. In *Proceedings of the 13th Workshop in Primary and Secondary Computing Education* (pp.1-4). https://doi.org/10.1145/3265757.3265770

Turan, S. B. (2022). İlköğretim matematik öğretmen adaylarının blok tabanlı programlamayı kullanma süreçleri ve tasarlanan öğrenme ortamlarının incelenmesi [The analysis of the block based programming processes and designed learning environments by prospective elementary mathematics teachers] (Doctoral thesis). Necmettin Erbakan Üniversitesi.

Üzümcü, Ö. & Bay, E. (2018). Eğitimde yeni 21. yüzyıl becerisi: Bilgi işlemsel düşünme [A new 21st century skill in education: Computational thinking]. *Uluslararası Türk Kültür Coğrafyasında Sosyal Bilimler dergisi 3* (2), 1-16.

Wing, M. J. (2006). Computational thinking. *Communications of The ACM 49*(3), 33-35. https://doi.org/10.1145/1118178.1118215

Wing, M. J. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of The Royal Society A* 366, 3717-3725. https://doi.org/10.1098/rsta.2008.0118

Wong, G. K. W., & Cheung, H. Y. (2020). Exploring children's perceptions of developing twenty-first century skills through computational thinking and programming. *Interactive Learning Environments*, *28*(4), 438-450.

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education (TOCE)*, *14*(1), 1-16.

Yıldız Durak, H. & Karaoğlan Yılmaz, F. G. (2019). Öğretmen adaylarının matematik öğretimine yönelik eğitsel dijital oyun tasarımlarının ve tasarım sürecine ilişkin görüşlerinin incelenmesi [An investigation of prospective teachers' educational digital game designs for mathematics teaching and their opinions on the design process]. *Ege Eğitim Dergisi*, *20*(1), 262-278. https://doi.org/10.12984/egeefd.439146

Yıldız, M., Çiftçi E & Karal, H. (2017). Bilişimsel düşünme ve programlama. A. İşman, H. F. Odabaşı ve B. Akkoyunlu (Ed.), Eğitim teknolojileri okumaları 2017 (ss. 75-86) içinde. Ankara: TOJET-Sakarya Üniveritesi.

Yükseköğretim Kurulu [YÖK] (2018). İlköğretim matematik öğretmenliği lisans programı. 21.12.2021,https://www.yok.gov.tr/Documents/Kurumsal/egitim_ogretim_dairesi/Yeni-OgretmenYetistirme-Lisans Programlari/Ilkogretim_Matematik_Lisans_Programi.pdf

Yükseltürk, E. & Üçgül, M. (2018). Blok tabanlı programlama. In Y. Gülbahar & H. Karal (Eds.), *Kuramdan uygulamaya programlama öğretimi* (pp. 2-36). Pegem Akademi.

Yünkül, E. Durak, G. Çankaya S. & Mısırlı, Z. A. (2017). Scratch yazılımının öğrencilerin bilgisayarca becerilerine etkisi [The Effects of Scratch Software on Students' Computational Thinking Skills]. *Necatibey Eğitim Fakültesi Elektronik fen ve Matematik Eitim Dergisi 11*(2), 502-517. https://doi.org/10.17522/balikesirnef.373424

GENİŞLETİLMİŞ ÖZET

1. GİRİŞ

Dünya birçok yönden hızlı bir değim içerisindedir fakat en belirgin değişim teknoloji alanında yaşanmaktadır. Değişen teknoloji ise, sivil hayatta, iş yerinde, evde olduğu gibi eğitim üzerinde de kapsamlı etkilere sahiptir (Allen & van der Velden, 2012). Özellikle son yıllarda yaşadığımız COVİD 19 salgın dönemi ile birlikte artık eğitim ve teknoloji kavramlarını ayrı ayrı düşünmek olanaksız hale gelmiştir (Mumcu, 2022). Günümüzde yaşanan hızlı değişimlerin üstesinden gelip evreni tanımak ve anlamlandırmak, sadece teknolojinin kullanıldığı alanları bilmekle değil; aynı zamanda bu değişimlere uyum sağlamaya yardımcı olacak becerilerin kazanılması ve geliştirilmesini de gerekli kılmaktadır (Erdoğan ve Şimşek, 2018). Bu anlamda teknolojideki hızlı gelişmelere paralel olarak eğitimciler, gelişen teknolojiyi geleneksel öğretim yöntemlerine başarılı bir şekilde entegre ederek öğretimi daha cazip kılma ve kolaylaştırma bağlamında önemli ilerlemeler sağlamışlardır (Baki, 2018). Günümüzde bireylerden problem çözme, eleştirel düşünme, analitik düşünme, yaratıcılık gibi 21.yy becerilerine sahip olması beklenmektedir ve bu becerilerin geliştirilmesinin en etkili yöntemlerinden biri de programlamadır (Yıldız vd., 2017). Programlama birçok fonksiyona sahiptir. Programlama öğrenerek insan, sistematik düşünme, problem çözebilme, olaylar arasındaki ilişkileri görebilme, yaratıcı düşünebilme gibi becerilere de daha kolay uyum sağlar. Kritik özelliklerinden dolayı, bu becerilerin küçük yaşlardan itibaren çocuklara kazandırılması önem kazanmaktadır (Gültepe, 2018). Söz konusu programlama olunca bu kavramdan ayrı olarak düşünülmeyen ve birbirini etkileyen diğer bir kavram da bilgi işlemsel düşünme becerisidir (Üzümcü ve Bay, 2018). Bilişim teknolojisinin her alanı kuşattığı ve sınırlarını her geçen gün genişlettiği günümüz koşullarında bilgi işlemsel düşünme sadece bilgisayar bilimcileri için değil geniş kitleler için gerekli olan temel bir beceridir (Wing, 2006, 2008). Bu açıdan öğretmen adaylarının bilgi işlemsel düşünme becerisinden uzak kalması söz konusu olmamalıdır. Bu haliyle öğretmen adaylarına kazandırılması gereken becerilerden biri de bilgi işlemsel düşünme becerisidir (Aydoğdu, 2020) ve programlama yoluyla bilgi işlemsel düşünme becerisinin geliştirilebileceği belirtilmektedir (Wong & Cheung, 2020). Ayrıca programlama faaliyetlerinin öğretmen adaylarının Teknolojik Pedagojik Alan Bilgisi (TPAB) gelişimleri üzerinde de etkisinin olduğu belirtilmektedir (Kim & Lee, 2018). Teknolojik pedagojik alan bilgisi (TPAB), öğretmenlerin dijital teknolojileri öğrenme araçları olarak entegre etme konusunda öğretimsel kararlar vermek için ihtiyaç duydukları bilgi hakkında düşünmeye yönelik bir çerçevedir (Niess, 2011). Programlama eğitiminin bilgisayar teknolojisine yönelik tutum üzerinde de etkisinin olduğu literatürde yer almaktadır (Erümit, 2020). Son zamanlarda programlama yapabilmeyi kolaylaştıran, kullanıcı dostu ve birçok görsel özellikler ile desteklenen Scratch, Google Blockly ve Code.org gibi açık kaynak platformlar programlama öğrenmeye yeni başlayan bireylere kılavuzluk etmek ve cesaretlendirmek için geliştirilmiştir (Aytekin vd., 2018).

Araştırmanın amacı Scratch programı eğitiminin ilköğretim matematik öğretmen adaylarının teknolojik pedagojik alan bilgileri, bilgi işlemsel düşünme becerileri ve bilgisayar bilimi tutumları üzerine etkisini incelemektir.

2. YÖNTEM

Araştırmada deneme modellerinden tek grup öntest-sontest deneysel desen kullanılmıştır. Araştırmanın çalışma grubunu bir devlet üniversitenin İlköğretim Matematik Öğretmenliği Lisans Programı IV. Yarıyılında (2. Sınıf) öğrenim gören Algoritma ve Programlama dersine kayıtlı 73 öğretmen adayı oluşturmaktadır. Öğretmen adaylarına Algoritma ve Programlama dersi kapsamında verilen Scratch programı eğitimin öncesi ve sonrasında; *Teknolojik Pedagojik Alan Bilgisi (TPAB) Ölçeği, Bilgi İşlemsel Düşünme Becerileri Ölçeği* ve *Bilgisayar Bilimi Tutum Ölçeği* uygulanarak ders kapsamında verilen eğitimin, öğretmen adayları üzerindeki etkisine bakılmıştır. Bilgisayar Bilimi Tutum Ölçeğinden alınan genel puanın ortalamaları arasındaki farkı tespit etmek için nonparametrik testlerden Wilcoxon işaretler sıralı testi kullanılmıştır. TPAB Ölçeği ve Bilgi İşlemsel Düşünme Beceri ölçeklerinden alınan genel puanın ortalamaları arasındaki farkı tespit etmek için ise parametrik testlerden bağımlı gruplar t-testi kullanılmıştır.

3. BULGULAR, TARTIŞMA VE SONUÇ

Araştırmada Teknolojik Pedagojik Alan Bilgisi (TPAB) ölçeğinin geneli ve alt boyutlarından elde edilen puanlar karşılaştırıldığında, ölçeğin genelinde puanlar arasında anlamlı bir farklılığı bulunmazken; ölçeğin sadece Alan Bilgisi (AB) ve Teknolojik Pedagojik Alan Bilgisi (TPAB) alt boyutlarında anlamlı bir farklılığın ortaya çıktığı görülmüştür. Öğretmen adaylarının süreç içerisinde Scratch projeleri geliştirirken ve ders planı hazırlarken seçilen konu-kazanımın alan bilgisi ile ilgili araştırma yapmaları, varsa eksiklerini görmeleri, gidermeleri ve matematiksel kavram ve bilgilerini Scratch programı ile ilişkilendirerek proje yapmaları alan bilgisi gelişimlerine katkı sağladığı söylenebilir. Ölçeğin TPAB alt boyutunda da anlamlı bir farklılığın ortaya çıktığı görülmüştür. Alanyazında da benzer sonuçlar görülmektedir. Dolayısıyla, Scratch kullanımının, uygun bir eğitim fırsatı bulmaları halinde öğretmen adaylarının teknolojik pedagojik alan bilgilerini geliştireceği söylenebilir.

Araştırmada Bilgi İşlemsel Düşünme Beceri Ölçeğinin geneli ve alt boyutlarından elde edilen puanlar karşılaştırıldığında puanlar arasında anlamlı bir farklılığın olmadığı görülmüştür. Aydoğdu (2020) tarafından yürütülen "Blok tabanlı programlama etkinliklerinin öğretmen adaylarının programlamaya ilişkin öz yeterlilik algılarına ve hesaplamalı düşünme becerilerine etkisi" adlı çalışmada, blok tabanlı programlama etkinliklerinin öğretmen adaylarının bilgi işlemsel düşünme becerileri üzerinde bir etkisinin olmadığı sonucuna ulaşıldığı görülmektedir. Söz konusu çalışmada bu sonucun, uygulama öncesinde öğretmen adaylarının ortalama puanlarının, ölçeğin orta noktasının üstünde olmasından kaynaklanmış olabileceği ifade edilmektedir. Bu araştırmada da başlangıçta öğretmen adaylarının ölçekten aldıkları puan ortalamalarının bilgi işlemsel düşünme becerisi ölçeğinin orta noktasının üstünde olduğu görülmektedir.

Araştırmada Bilgisayar Bilimi Tutum Ölçeğinden elde edilen puanlar karşılaştırıldığında anlamlı bir fark olmadığı sonucuna ulaşılmıştır. Köseoğlu vd. (2007) biyoloji öğretmen adayları ile 10 hafta süreyle yürütmüş oldukları çalışmada verilen bilgisayar kursunun öğretmen adaylarının bilgisayara yönelik tutumları üzerinde olumlu bir etkisinin olmadığı sonucuna ulaşmışlar ve bu sonucu kurs süresinin tutum geliştirmeye yeterli olmadığı şeklinde değerlendirmişlerdir. Bu araştırmada da bir dönem boyunca yapılan eğitim ve uygulamalarının öğretmen adaylarında bilgisayar bilimine yönelik tutumlarını geliştirmek için yeterli bir süre olmadığı düşünülebilir. Akçay ve Çoklar (2016) programlama becerilerinin üretim merkezli beceriler olup üretimin vermiş olduğu güdülenme ile olumlu sonuçlar ortaya koyacağını ifade etmektedirler. Öğretmen adaylarının bilgisayar bilimine yönelik tutumlarında anlamlı bir farkın çıkmamasının bir başka sebebi olarak, sürecin programlama becerilerini geliştirmede yeterli olmaması olarak düşünülebilir.

Sonuç olarak, geleneksel öğretim anlayışının terk edildiği günümüzde, öğretim teknolojilerinde yaşanan değişim ve dönüşümün etkisiyle, gelişen birçok öğrenme ve öğretme yaklaşımından ve bunların eğitim süreçlerine etkisi kaçınılmazdır (Mumcu, 2022). Eğitim alanındaki birçok yeniliğin öğrenme ve öğretme süreçlerinde kullanılmasında, öğretmenler en kritik role sahip paydaşlar olarak görülmektedir (Admiraal vd., 2017). Eğitim sisteminde olası değişimlerin başarısında öğretmenlerin yeniliklere açık olması ve uyum sağlayarak gerekli beceri kazanması ile doğrudan ilgilidir (Fullan, 1991). Aynı zamanda teknolojinin öğrenme öğretme sürecinde başarılı şekilde kullanılması öğretmenlerin teknolojiyi kabul etmeleri ve kullanma niyetleriyle de ilişkilidir (Admiraal vd., 2017). 2024 ortaokul matematik dersi öğretim programı içerisinde algoritma, akış şeması, programlama, kodlama kavramlarına sıklıkla yer verilmiş olması programlamanın müfredata entegre edilmesi yönündeki çalışmaları da ortaya koymaktadır. Bu anlamda Scratch programlama aracı kullanılarak yürütülen bu çalışma dışında farklı programlama araçları kullanılarak çalışmaların yapılması geleceğin müfredat uygulayıcıları olacak olan öğretmen adayları için faydalı olacaktır.

ARAŞTIRMANIN ETİK İZNİ

Bu çalışmada "Yükseköğretim Kurumları Bilimsel Araştırma ve Yayın Etiği Yönergesi" kapsamında uyulması gerektiği belirtilen tüm kurallara uyulmuştur. Yönergenin ikinci bölümü olan "Bilimsel Araştırma ve Yayın Etiğine Aykırı Eylemler" başlığı altında belirtilen eylemlerden hiçbiri gerçekleştirilmemiştir.

Etik kurul izin bilgileri

Etik değerlendirmeyi yapan kurul adı: Necmettin Erbakan Üniversitesi Sosyal ve Beşeri Bilimler Bilimsel Araştırmalar Etik Kurulu

Etik değerlendirme kararının tarihi: 19.02.2021

Etik değerlendirme belgesi sayı numarası: 2021/83

ARAŞTIRMACILARIN KATKI ORANI

Araştırmacılar çalışmaya eşit oranda (%50) katkı sağlamıştır.

ÇATIŞMA BEYANI

Bu çalışma Necmettin Erbakan Üniversitesinde yürütülen 1. yazarın doktora tez çalışmasından üretilmiştir. Çalışma üzerinde herhangi bir çıkar çatışması bulunmamaktadır.