CEMRE KÖKSALOĞLU

2022

BLOCK-BASED CODING IN K-12 EDUCATION: A SYSTEMATIC LITERATURE REVIEW

CEMRE KÖKSALOĞLU

A THESIS SUBMITTED

FOR

THE DEGREE OF MASTER OF ARTS

IN

CURRICULUM AND INSTRUCTION

İHSAN DOĞRAMACI BILKENT UNIVERSITY

ANKARA

FEBRUARY 2022

Dedicated to my loved ones

Block-Based Coding in K-12 Education: A Systematic Literature Review

The Graduate School of Education

of

İhsan Doğramacı Bilkent University

by

Cemre Köksaloğlu

In Partial Fulfilment of the Requirements for the Degree of

Master of Arts

in

Curriculum and Instruction

Ankara

February 2022

İHSAN DOĞRAMACI BILKENT UNIVERSITY

GRADUATE SCHOOL OF EDUCATION

Thesis Title: Block-Based Coding in K-12 Education: A Systematic Literature Review

Cemre Köksaloğlu

January 2022

I certify that I have read this thesis and have found that it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Arts in Curriculum & Instruction.

Asst. Prof. Dr. Armağan Ateşkan (Advisor)

I certify that I have read this thesis and have found that it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Arts in Curriculum & Instruction.

Assoc. Prof. Dr. Erdat Çataloğlu (Examining Committee Member)

I certify that I have read this thesis and have found that it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Arts in Curriculum & Instruction.

Prof. Dr. Arif Altun, Hacettepe University (Examining Committee Member)

Approval of the Graduate School of Education

Prof. Dr. Orhan Arıkan (Director)

ABSTRACT

BLOCK-BASED CODING IN K-12 EDUCATION: A SYSTEMATIC LITERATURE REVIEW

Cemre Köksaloğlu

M.A., Program of Curriculum and Instruction

Advisor: Asst. Prof. Dr. Armağan Ateşkan

February 2022

This study aims to determine the methodological tendencies and outputs of the studies conducted on the use of block-based coding tools in K-12 education by systematic literature review. According to their demographic characteristics, a total of 59 full-text research articles were analyzed through thematic content analysis. In addition, learning domains were explored in the research articles, and results were revealed by themes regarding the impact of block-based coding in K-12 education on learning domains. Research findings display that most of the research articles were published in 2019, commonly using the mixed-method research, and they were carried out with middle school level students with less than 50 sample sizes. This review also found that most research studies were conducted in the United States and in the Computer Science subject area. Finally, the most frequently used block-based coding tool was Scratch and problem-based learning approach was the most used considering the teaching/learning approaches. Besides, the cognitive domain was explored mainly in the research articles. The results of the research articles indicated that block-based coding tools improve the following skills regarding cognitive abilities: computational thinking, problem-solving, algorithmic and reflective thinking, critical thinking, and creative thinking skills. Moreover, results for interdisciplinary learning and gender differences regarding the cognitive domain were also found. Additionally, block-based coding positively affected students' feelings regarding the affective domain. Lastly, whereas students had positive attitudes towards coding, negative attitudes were also reported.

Keywords: block-based coding, technology, education, thematic content analysis, K-12, learning domains

ÖZET

K-12 Eğitiminde Blok Tabanlı Kodlama: Sistematik Literatür İncelenmesi

Cemre Köksaloğlu

Yüksek Lisans, Eğitim Programları ve Öğretim

Danışman: Dr. Öğr. Üyesi Armağan Ateşkan

Şubat 2022

Bu çalışma, blok tabanlı kodlama araçlarının K-12 eğitiminde kullanımına yönelik yapılan çalışmaların metodolojik eğilimlerini ve çıktılarını ana hatlarıyla belirlemeyi, kapsamlı ve bütüncül olarak incelemeyi amaçlamaktadır. Bu sistematik literatür incelemesinde Web of Science ve Scopus veri tabanları kullanılarak çalışmanın amaçlarına ulaşılmıştır. Toplam 59 araştırma makalesi demografik özelliklerine göre tematik içerik analizi ile analiz edilmiştir. Ayrıca, araştırma makalelerinde öğrenme alanları araştırılmış ve A-12 eğitiminde öğrenme alanlarının blok tabanlı kodlamaya etkisine ilişkin temalarla sonuçlar ortaya konulmuştur. Araştırma bulguları; araştırma makalelerinin çoğunun 2019 yılında yayınlandığını, en çok tercih edilen araştırma yönteminin karma yöntem olduğunu ve örneklem büyüklüğü 50'den az olan ortaokul düzeyindeki öğrencilerle gerçekleştirildiğini göstermektedir. Bu inceleme aynı zamanda çoğu araştırmanın Amerika Birleşik Devletleri'nde ve Bilgisayar Bilimi alanında yapıldığını da ortaya koymuştur. Son olarak derslerde en sık kullanılan blok tabanlı kodlama aracının Scratch olduğu bulunmuştur ve çoğunlukla problem çözmeye dayalı öğretme/öğrenme yaklaşımı tercih edilmiştir. Demografik özelliklerin yanı sıra, ağırlıklı olarak bilişsel alanda çalışmalar tespit edilmiştir. Araştırma makalelerinin sonuçları, blok tabanlı kodlama araçlarının bilgi işlemsel düşünme, problem çözme, algoritmik ve yansıtıcı düşünme, eleştirel düşünme ve yaratıcı düşünme becerileri gibi bilişsel yeteneklerle ilgili bazı becerileri geliştirdiğini göstermiştir. Ayrıca bilişsel alana ilişkin disiplinler arası öğrenme ve cinsiyet farklılıklarına ilişkin sonuçlar da bulunmuştur. Bunlara ek olarak blok tabanlı kodlama, öğrencilerin duyuşsal alana ilişkin duygularını da olumlu yönde etkilemiştir. Son olarak öğrencilerin kodlamaya yönelik tutumlarında hem olumlu hem de olumsuz bulgulara rastlanmıştır.

Anahtar kelimeler: blok tabanlı kodlama, teknoloji, eğitim, tematik içerik analizi, A-12, öğrenme alanları

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest appreciation to my advisor Asst. Prof. Armağan Ateşkan for her patience, assistance, invaluable advice, continuous support, insightful comments and suggestions during this study.

I would also like to extend my sincere gratitude to the members of my thesis committee, Assoc. Prof. Erdat Çataloğlu and Prof. Dr. Arif Altun, for sparing their valuable time to read and advise my thesis, listen to my presentation, and to enlighten me with their suggestions to improve this study.

I am also grateful to each member of my family starting with my dear mother Uğur Köksaloğlu, my dear father Mehmet Köksaloğlu, my little and lovely brother Efe Köksaloğlu, my dearie grandmother and beloved uncle Ufuk Pekparlak. The completion of my thesis would not have been possible without the endless support of them.

Finally, I would like to thank my valuable friends and colleagues, especially Selin İnal and Hüsna Dolma for their contribution and help for this study. Also, many special thanks to Fatih Ceyhanlı for his profound belief in my study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

Introduction

Education has a significant role in each person's life, considering the fact that being a well-educated person opens a variety of opportunities all the time. In our contemporary world, being a scholar person and being a technology-literate person is crucial. Technology has been a central aspect of our daily lives since numerous technological tools surround us. Thus, individuals should get this competence from early childhood. Bers et al. (2019) recognized that today's children are already being raised with smart devices. That is to say, the society is expected to understand technology better gradually and become a technology producer for the future workforce (Yildiz et al., 2020).

In this digital era, technology has an important role in education (Akkoyunlu, 2002). Lately, initiatives to improve educational systems for raising individuals with twenty-first-century skills are increasing. An example of those initiatives is the change of contemporary curricula to include topics considered in today's world, such as coding at schools as a new literacy (Bers et al., 2019).

The study aims to classify the demographic characteristics, which are the year of the publications, research methodologies, data collection instruments, the grade level of the study sample, sampling sizes, countries research carried out, subject areas covered in the articles, block-based coding tools, and teaching/learning approaches. Another goal of this study is to explore the learning domains regarding outputs of the scholarly articles in the literature conducted on the use of block-based coding in K-12 education. In this regard, the background of the study, the statement of the problem, the purpose, the research questions, the significance of the study, the limitations, and the definition of the essential terminologies are all included in this chapter.

Background

Computer Science education is spread worldwide, and coding has become a part of compulsory education in several countries. In the report of European Schoolnet (2015), 2014 was the year of coding integration into the curriculum, and England has attempted first. At first, coding education is compulsory in England's primary, middle, and high school levels. It is the first country to include "Computer Science" as a mandatory subject in its curriculum for children aged 5-16. In addition to England, the report has analyzed how ministries of education integrated coding into their curricula. The results showed that 16 European countries, Austria, Bulgaria, The Czech Republic, Denmark, Estonia, France, Hungary, Ireland, Israel, Lithuania, Malta, Spain, Poland, Portugal, Slovakia, and England have integrated coding into their curricula. Besides, while Finland, Belgium, the USA, and South Korea have been planning, the Netherlands and Norway have had no plans to integrate it. Also, Turkey has recognized the importance of coding, and coding education has incorporated into the Turkish curriculum in the 2012-2013 academic year (Demirbaş, 2019).

In Turkey, the Ministry of National Education (MoNE) constituted an educational content network called Education Information Network (EBA) which is designed and run by the Innovation and Educational Technologies General Directorate. This network aims to integrate technology into education when needed, and it provides teachers and students with online access to course materials as part of the FATIH project. In this regard, a block-based coding tool called “MASKOD” in

EBA platform was created in order to acquire coding skills by students within the scope of the 2023 education vision (Ministry of National Education [MoNE], 2019; Organisation for Economic Co-operation and Development [OECD], 2020).

Activities also support Computer Science education outside of the school in the last decades with the help of several global events, including Europe Code Week, which volunteers have run since 2013. This activity aims to encourage students to learn the basics of coding and computational thinking. According to the statistics, 4.2 million students attended the Europe Code Week in more than 80 countries worldwide in 2019. While the average age of participants was 11, the participants consisted of 49% females (CodeWeek, n.d). In addition, Hour of Code activities is done worldwide by Code.org, which is a block-based coding tool (Code.org, n.d; Weintrop, 2019). The Hour of Code has been presented in the scope of Computer Science education in the second week of December annually (Code.org, n.d.). The activity's primary goal is to make students interested in computer science and programming.

Digital Fluency

In the twenty-first century, the concept of digital fluency is to be broadened (Resnick, 2012) to incorporate designing and creating content, not just browsing and interacting with them. Digital fluency is much more about expressing, exploring, and realizing the ideas than the capability of using technological tools (Resnick & Rusk, 1996). Hence, there are increasing initiatives to society's well-being by raising individuals with twenty-first-century skills such as understanding significant mathematical and computational terms, thinking creatively, reasoning systematically, and working collaboratively (Resnick et al., 2009). In this context, a new literacy, coding, is becoming an essential skill in this digital era (Polyviou, 2021).

Bloom’s Taxonomy

Sobral (2021) stated that Bloom's taxonomy can be considered to teach or learn in the introduction to coding. According to Bloom's taxonomy (Bloom et al., 1984), there are three learning domains: cognitive, affective, and psychomotor (knowledge, skills, and attitudes, respectively). In this respect, coding teaches computational thinking skills to improve higher-order cognitive domain skills such as problem-solving, analytical and logical thinking, and reasoning (Webb et al., 2016). Furthermore, coding education can also be considered in the affective domain regarding Bloom's taxonomy, as Sobral (2021) stated. Since block-based coding tools provide students with a constructive environment, they could have positive feelings and attitudes towards coding.

Computational Thinking

According to Wing (2006), computational thinking is a prerequisite for each child's analytical thinking abilities, such as designing, problem-solving, and enabling them to adapt and improve strategies. Therefore, educational institutions aim at improving students from kindergarten through high school by increasing their computational thinking abilities. At this point, there are varied terms used for coding like “programming,” “computing,” “computer programming,” “computer science,” and “computational thinking” to refer to the integration of coding skills into the curriculum (Armoni, 2016). According to Temurnikar (2020), coding has importance in twenty-first-century education, such as preparing for the new age reality, nurturing problem-solving qualities, enabling brain development, developing creative skills, and building strong characters.

Coding Tools

Furthermore, Armoni (2016) also stated that some educational environments in coding were developed and can be used for K-12 education. To be more specific, there are two types of coding environments: text-based and block-based (Price & Barnes, 2015). Weintrop (2019) states that block-based coding environments provide learners with visual clues for how and where commands may be employed, which makes it easier to acquire coding skills. In contrast, text-based coding can be a complex way to learn programming at first (Codejig, n.d.). Therefore, many user-friendly block-based coding tools have been developed in recent years, such as Scratch, Code.org, Alice, etc. These tools do not require any syntax, and they are more straightforward for all grade levels to learn the coding logic effectively (Weintrop, 2019).

In this context, Noone and Mooney (2018) conducted a systematic literature review about the comparison of block-based programming language and text-based programming language regarding educational benefits on the learning process. They put forward the idea that block-based programming languages have several advances compared to text-based programming languages in terms of being accessible since there are several coding tools and user-friendly. They also found that textual programming is better understood by high school students. More specifically, the block-based programming language is found to be the most beneficial for students under the age of 14, and a hybrid environment, which includes both block-based and text-based elements simultaneously, would be the most effective for those around the age of 15–17.

All in all, CS education is available globally both at school and outside of the school context. Also, as a part of CS education, coding is essential in order to improve students’ thinking styles in terms of learning domains. This study is based on block-based coding tools. For, although there are researches done about coding, there is no study about block-based coding in K-12 education, specifically by systematic literature review.

Problem

In today's world, technology integration in education is inevitable nowadays, and using twenty-first-century skills is significant, especially in education, because of the needs of society regarding future employment. Therefore, educational institutions should follow the rapidly changing educational tools continuously. One of the examples of these tools is teaching coding to students as a new literacy from kindergarten through high school. In that respect, educational institutions have started integrating this prominent tool into their ICT curriculum by implementing coding education.

Coding, a new literacy, is the process of “giving instructions to computers on what actions to perform” using computer programming languages (Weinstein, 2021). As stated before, programming languages have two types: text-based and block-based (Price & Barnes, 2015). Block-based coding is a fairly new topic in education. Research studies in the literature about implementing block-based coding tools in K-12 education from many perspectives are investigated. To illustrate, Talan (2020) conducted a meta-thematic analysis to investigate the common research methodologies and the results of the studies conducted on using one of the block-based coding tools, Scratch software, in education. This study indicated that using Scratch as a tool has provided improvements regarding both cognitive and affective processes such as higher motivation, self-efficacy, attitude, higher-level thinking, and academic success of students.

However, if the literature is inspected, there is no thematic content analysis done about block-based coding in K-12 education. To explain further, there is a need to draw a general framework on block-based coding in K-12 education to reveal the demographic characteristics within the research articles. Additionally, Strawhacker and Bers (2018) stated that even if this new literacy has grown in popularity among educational institutions, there is not sufficient information on what learning domain and which skills students use as they code. Hence, this research aims to fill the gaps in the literature by exploring which learning domains are explored in the research studies and results of those studies regarding the impact of block-based coding in K-12 education on learning domains.

Purpose

The purpose of this study is to review research articles published in the time frame between 2000 and 2020 that were conducted in the area of block-based coding at the K-12 level according to their demographic characteristics, including years, research methodologies, data collection instruments, the grade level of the sample, and sampling sizes. Moreover, the distribution of countries, subject areas, block-based coding tools, and teaching/learning approaches are also categorized in this study. In addition, this study aims to explore which learning domains are explored in the research articles. Another aim is to reveal the results of research articles regarding the impact of block-based coding in K-12 education on learning domains. This research reports the literature in an organized way, identifies gaps, and indicates the commonly used ones in results by gathering information from research articles. In this regard, 59 research articles were analyzed through complete examination.

Research Questions

This study addresses the following main and sub-questions:

1. What are the demographic characteristics of the research articles in the literature about block-based coding in K-12 education in terms of;
   a. year of the publications
   b. research methodologies
   c. data collection instruments
   d. the grade level of the sample
   e. sampling sizes
   f. countries
   g. subject areas
   h. block-based coding tools
   i. teaching/learning approaches
2. How are learning domains explored in the articles about block-based coding in K-12 education?
   a. Which learning domains are studied and/or explored in the articles about block-based coding in K-12 education?
   b. What are the results of those studies regarding the impact of block-based coding in K-12 education on learning domains?

Significance

This study attempts to classify and analyze research articles' methodological tendencies and results on the use of block-based coding tools in K-12 education over the past 20 years comprehensively and holistically. Conducting such research is critical for improving the efficiency and applicability of block-based coding tools. The number of articles in the associated literature relating to the use of block-based coding tools in education has expanded significantly, attracting many researchers' attention. It is expected that the findings of this study add a new dimension to future studies in the subject and pave the way for further research.

The significance of this study is to have an idea about what has been done so far about block-based coding in K-12 education by reviewing the literature systematically. The findings of this study provide information to reveal demographic characteristics of the research articles in the literature about block-based coding in K-12 education. In addition, this study aims to find which learning domains are explored in the articles about block-based coding in K-12 education and the results of those studies regarding the impact of block-based coding in K-12 education on learning domains.

As a result of this study, collecting and reviewing the research articles provides insights about block-based coding in K-12 education for educators, teachers, and other researchers interested in that area. Also, reviewed literature helps to find gaps in the research that could be considered for future research and identifies the commonly used block-based coding tools in results.

Limitations

This research is limited to scientific studies about block-based coding in K-12 education between the years 2000 and 2020. Additionally, this study is limited to full-text access to only research articles published in refereed journals in English languages. It does not include research articles conducted in different languages and other publications such as book chapters, papers, book reviews, and opinion essays.

Definition of Key Terms

Block-based coding: It is a learning environment that includes dragging and dropping coding blocks to create animated games and stories (Barone, 2020).

Computational thinking (CT): It is an essential skill including the comprehension of computer science concepts, the ability to solve problems, design systems, and understand human behavior (Wing, 2006).

K-12 education: 12 years from kindergarten to primary and secondary education.

Twenty-first-century learning skills: They are “learning and innovation skills, information, media, and technology skills, and life and career skills”. Learning and innovation skills are composed of “4C's: creativity, critical thinking, problem-solving, communication, and collaboration” (Battelle for Kids, 2019).

CHAPTER 2: REVIEW OF RELATED LITERATURE

Introduction

In our ever-evolving world, having twenty-first-century skills is significant for students' education in a society that expects high-level adaptability to this world. Twenty-first-century skills are comprised of “learning and innovation skills, information, media, and technology skills, and life and career skills” (Battelle for Kids, 2019). Therefore, students should have these skills regarding analyzing and solving problems critically and thinking creatively.

Recently, technology, which can be used to improve twenty-first-century skills, has been an indispensable part of education. Therefore, various technological tools are being used in teaching and learning in the context of educational technology (Ganimian et al., 2020). According to “the Association for Educational Communications and Technology (AECT)” educational technology can be defined as “the study and ethical application of theory, research, and best practices to advance knowledge as well as mediate and improve learning and performance through the strategic design, management and implementation of learning and instructional processes and resources” (Januszewski & Molenda, 2008, p.1). Johnson et al. (2014) also defined educational technology as “in a broad sense as tools and resources that are used to improve teaching, learning, and creative inquiry” (p.32). There are plenty of innovations regarding educational technologies such as coding that aim to develop those skills. Coding, a new kind of literacy for the twenty-first century (Polyviou, 2021), is defined as giving commands to execute specific tasks (Tynker, 2021).

This chapter aims to share the necessary background information about computer science, computer science education, future implications, and the ways of improving computer science education. Following this, the theoretical background of learning domains, computer science education regarding learning theories, learning approaches, learning ideologies, and the general framework of the computer science curriculum are discussed. This chapter also explains computational thinking and visual and block-based programming languages.

Computer Science

Tucker et al. (2003) defines Computer Science (CS) as "the study of computers and algorithmic processes, including their principles, hardware and software designs, implementation, and their impact on society" (p.6). What is more, Brookshear (2006) described the CS as:

> Computer science (CS) is the discipline that seeks to build a scientific foundation for such topics as computer design, computer programming, information processing, algorithmic solutions of problems, and the algorithmic process itself. It provides the underpinnings for today's computer applications and the foundations for tomorrow's computing infrastructure. (p.2)

In brief, Barr and Stephenson (2011) summarized the principles of CS as "programming, hardware design, networks, graphics, databases and information retrieval, computer security, software design, programming languages and paradigms, logic, translation between levels of abstraction, artificial intelligence, the limits of computations, applications in information technology and information systems, and social issues" (p.49).

Computer Science Education

CS education is crucial in today's modern and digital society. Considering the K–12 Computer Science Framework (2016), CS education is based on “computer literacy, educational technology, digital citizenship, and information technology” (p.13). Initially, computer literacy is about the use of computers and their applications. Secondly, educational technology is about applying computer literacy to school subjects. Thirdly, digital citizenship is about using technology regarding appropriateness and responsibilities. Lastly, information technology is focused on the background of computer science.

The recognition of CS education's significance has been expanded worldwide, and it is included in the curriculum of not only primary, but also secondary schools, providing options in terms of curriculum requirements (Vegas et al., 2021). Vegas and Fowler (2020) also found that 44 countries (around 20 percent) out of 219 offer CS education as a compulsory or elective course. In addition, 15 countries (approximately 7 percent) offer this education as an elective course in some selected schools like states and provinces. And then, 160 countries (around 73 percent) offer CS education as a pilot program. In this regard, Vegas and colleagues (2021) investigated which nations worldwide have included CS in their school curriculum for both primary and secondary education. It has been demonstrated that while some of the countries in Europe and Asia offer CS education as a mandatory course, it is provided as an elective course at some states or provinces in U.S. and Canada. In addition, most countries in the Western European continent offer CS education in an interdisciplinary way. Many countries in Latin America and Central and Southeast Asia have CS education programs and projects at the K-12 level. However, it was seen that in Africa and the Middle East countries in which CS education is not integrated into the curriculum sufficiently. Additionally, how CS education varied by region and income level to explore ways to improve CS education was also reported by Vegas and colleagues (2021). They found that CS education is provided more in countries with higher incomes.

One of the issues regarding computer science education in the twenty-first century is gender differences, which is a part of scientific research on this subject (Beyer, 2014). There is an initiative called Computer Science for All (CSA) to highlight the importance of computing and the need for all students to get an opportunity to have this education (CSforAll, n.d). However, there exist some issues regarding equitable learning environments in CS education, such as gender diversity. Accordingly, Miliszewska et al. (2006) demonstrated internal and external factors regarding the lack of female participation in computer science courses. They pointed out that external factors caused to be prejudiced of females towards the computing courses, such as the perception of male domination in computing culture. Therefore, females lose their self-confidence and computational abilities. On the other hand, internal factors also affect females' perceptions against these courses negatively, such as “negative classroom experiences, inadequate equipment, unequal participation in class, lack of gender-neutral projects in the course, lack of role models and career advice, insufficient institutional support, and lack of peer support groups” (p. 109-110).

Likewise, teachers in the study of Gretter et al., (2019) stated that they had difficulties incorporating female students in taking computer science classes because of gender stereotypes in society regarding computing. In a similar context, Master et al., (2016) explored how the female students’ interests in computer science courses were affected depending on the classroom environment. The results showed that putting the stereotypical male-dominated items including video games, computer parts, electronics in the learning environment influenced females' interest in CS courses negatively, whereas putting the non-stereotypical objects such as art, nature, pen, coffee pictures were not. What is more, Taylor (2016) stated that female students are less aware, less involved, and less confident in learning CS than male students. Even though there exist some issues regarding inequalities in CS education, there could be some improvements in the curriculum.

Future Implications and The Ways of Improving Computer Science Education

In many countries, there are some attempts to provide CS sources in the curricula through changes in the learning environment. In this context, Gretter and colleagues (2019) further examined two barriers: self-identity and isolation in computing activities. Teachers in the study shared methods to retain students in computer science education, such as creating an inclusive classroom environment and increasing a sense of belonging. Teachers could create a more welcoming classroom atmosphere for females and males regarding physical artifacts.

It is also required to raise awareness of female students towards computer science education. Hence, Google CS Ed Research Group (2014) conducted a study to identify the factors that affect a female's interest in computer science careers. According to the study, these factors are “social encouragement, self-perception, academic exposure, and career perception” (p.2). That is to say, there needs to have positive reinforcement from family, peers, and educators, provide opportunities for attending courses and activities and practicing computational skills, and have role models from the field. In short, these ways can be integrated into the curriculum.

Lastly, Tissenbaum and Ottenbreit-Leftwich (2020) also stated their viewpoints for the Vision of K–12 Computer Science Education for 2030 that the computer science curriculum should be “more student, community, and equity centered” (p.42).

Theoretical Background of Learning Domains

Benjamin S. Bloom, an American educational psychologist, formulated a Taxonomy of Learning Domains system in 1956. This taxonomy includes three domains that are Cognitive, Affective, and Psychomotor Domains. Initially, the Cognitive Domain includes the development of mental skills and information processing. Secondly, the Affective Domain comprises the ways that are associated with the things such as “emotions, feelings, values, appreciation, enthusiasms, motivations, and attitudes” (Krathwohl et al., 1973). Lastly, the Psychomotor Domain represents “physical movements, coordination, and use of the motor-skill areas” (Simpson, 1972).

Furthermore, each domain has its categories. According to Bloom’s revised taxonomy by Anderson and Krathwohl (2001), who modernized Bloom’s original taxonomy according to the needs of the twenty-first-century education, the cognitive domain of Bloom’s revised taxonomy consists of six levels from simplest to more advanced thinking skills: “remembering, understanding, applying, analyzing, evaluating, and creating” (p.28). Besides, under the cognitive domain, skills regarding cognitive abilities also include computational thinking skills. Computational thinking skills comprise problem-solving, algorithmic and reflective-thinking, critical-thinking, and creative-thinking skills (Wing, 2006).

The affective domain includes five levels: “receiving, responding, valuing, organizing, and characterizing” (Krathwohl et al., 1973). According to Hewitt et al. (n.d), the affective domain also has skills for each level which are “being open to experience (receiving), engaging in life (responding), cultivating values (valuing), managing oneself (organizing), and developing oneself (characterizing)" (p.214).

The psychomotor domain has seven levels from the most straightforward behavior to the most complicated: "perception, set, guided response, mechanism, complex overt response, adaptation, and origination" (Simpson, 1972).

One of the studies investigated the outputs of the research articles regarding the use of the Scratch tool in education (Talan, 2020). According to the findings of this study, some variables were classified in terms of cognitive processes, affective processes, and learning environment/system. As a result, the learning level/success regarding cognitive processes was the most investigated variable. The second and third most investigated variables were problem-solving skills and self-efficacy. Moreover, in terms of affective processes, attitudes, motivation, and interest variables were predominant in the previous research studies. Lastly, Talan (2020) analyzed the effects of cooperative work, active participation, and system availability variables under the learning environment/system title.

Computer Science Education Regarding Learning Theories, Learning Approaches, and Learning Ideologies

According to the New Media Consortium Horizon Report for K-12 education (Johnson et al., 2014), "project-based learning, problem-based learning, inquiry-based learning, and challenge-based learning" (p.8) and similar learning approaches are student-centered, and therefore beneficial in active learning (Johnson et al., 2014; Westwood, 2008). One of the most popular and widely used methods is project and problem-based in CS education (Garneli et al., 2015). Sáez-López et al. (2016) also claimed that there are several benefits of project-based learning as a constructivist approach to improving in problem-solving strategies and discovering subjects.

Moreover, students can have a chance to develop themselves with the help of learning by doing and creating. Similarly, CS education can be related to the learner-centered ideology from Schiro's (2013) curricular ideologies. Learner-centered ideology aims to adapt education regarding the needs and concerns of students.

On the other hand, CS Education can also be linked with the social-efficiency ideology. For the social-efficiency ideology, the main purpose of education is to equip individuals with skills for the needs of the workplace and society (Schiro, 2013). Therefore, getting a CS education is vital for students because there is an increasing demand for computing jobs in the job market (Alexiou-Ray et al., 2020).

Besides learning approaches and ideologies, CS education is focused on the constructionism learning theory with regards to teaching and learning (Machanick, 2007). Wulf (2005) also stated that constructivism is based on student-centered pedagogy emphasizing learning through exploration. That is to say, students construct their knowledge rather than acquiring it through textbooks or lectures (Ben-Ari, 1998). According to Vygotsky's social constructivism approach, the Zone of Proximal Development (ZPD) concept has its place in CS education regarding problem-solving abilities. For instance, online learning environments where students can learn coding with drag-and-drop methods act as a scaffolding. In other words, these online coding environments make students develop their knowledge and problem-solving skills by exploring (Rose et al., 2020) and prevent students from making syntax errors compared to text-based coding environments (Sentence et al., 2019). That is to say, block-based coding settings allow learners to think creatively, think critically, solve problems even in real-life contexts, and work collaboratively by engaging with a subject.

General Framework of Computer Science Curriculum

Since CS education is based on the constructivist approach, the CS curriculum can be related to a postmodern curriculum that provides students with a learning environment where learners explore, discuss, and interpret the topics or issues (Lau, 2001). Furthermore, CS education can be associated with the curriculum as a process and a praxis model. In the curriculum as a process model, teachers and students communicate in order to explain and analyze information. Besides the process model, the praxis model is based on information and action (Smith, 2000). For instance, there are some latest trends in the CS curriculum, such as teaching coding to students. Coding is one of the dimensions of CS education as part of a curriculum (Alexiou-Ray et al., 2020) since it is a new literacy and is becoming an essential skill in this digital era. Resnick et al. (2009) have pointed out that there is a place in the CS curriculum to teach children how to code in the late 1970s and 1980s.

Moreover, coding is related to the "International Society for Technology in Education (ISTE) Standards" (ISTE, 2016). According to the ISTE Standards, students should have seven necessary skills: "Empowered Learner, Digital Citizen, Knowledge Constructor, Innovative Designer, Computational Thinker, Creative Communicator, and Global Collaborator" in the scope of the CS curriculum (ISTE, 2016). These standards have been adopted all around the world.

According to the Vision for K-12 Computer Science Framework (2016), the CS curriculum aims to raise informed citizens with "problem-solving, computational thinking, and creative thinking skills" (p.10). This framework also indicated that students could have discussions about computer science subjects using critical thinking skills. Also, students will develop skills of computer science in terms of knowledge and creating while they become better users. Besides, students can further comprehend the concept of computer science in the real-world, and therefore they can “learn, perform, and express themselves in other subjects and interests” (p.10).

In addition to the general framework of the CS curriculum, there are basic principles of the Turkish National Education System. The Turkish National Education System's primary purpose is to raise individuals with self-confidence, self-discipline, who have the ability of learning and improving their aptitude, and support their physical, mental, and emotional development (MoNE, 2018).

Moreover, computer science education is given under the curriculum's Information and Communication Technologies (ICT) course. According to the country report on ICT in education in Turkey by European Schoolnet (2017), the Ministry of Development (MoD) developed a plan called "Information Society and Action Plan (ISAP)" regarding digital education policy. The fundamental purpose of this development plan was to "educate individuals to obtain fundamental skills necessary for the information society" (p.6). In this context, there are also some digital education initiatives. One of those initiatives is the implementation of computing, coding, and computational thinking initiatives. According to this initiative, the ICT course was modified to include coding education. It has become a mandatory course for students in grades 5 and 6 and an elective course for grades 7 and 8. Besides computer literacy, coding education aims to provide all students with algorithmic thinking skills. Coding abilities will aid students in comprehending programming logic by allowing them to create a little piece of computer software. Students who learn to code will be the candidates of the future software-developers.

Furthermore, there are primary objectives of the ICT course for K-12 education. The main objectives of the ICT course for primary education stated by MoNE (2018a) are gaining awareness of the correct and effective use of information technologies; using technology ethically and safely; creating an idea of the concept of technology and research purposes; creating products using information technologies; acquiring problem-solving and computational thinking skills and developing an understanding of algorithm design; using different logic structures to solve problems and doing the programming by designing their games.

The main objectives of the ICT course for secondary education stated by MoNE (2018b) include being a digital citizen who understands technological concepts, systems and processes well. Another objective is to use information technologies effectively and appropriately. In this regard, accessing, researching, and using internet-based services are parts of the main objectives. Additionally, setting the basis of computer science knowledge, and acquiring and developing problem-solving and computational thinking skills are included. Furthermore, students are expected to follow and evaluate the reasoning process and acquire cooperative working skills. Similarly, not only benefiting from social environments and sharing what they learn as part of the learning process but also seeking learning opportunities online are incorporated in the MoNE ICT curriculum. Other objectives comprise of expressing linguistically and artistically by creating an idea of the algorithm design; finding solutions to problems through the application of the right programming approach; having further technical skills of coding language. Likewise, being able to efficiently employ one coding language, carrying out studies on product design and management; providing solutions to daily life issues through creative and authentic projects, and gaining awareness about lifelong learning are included in the MoNE ICT curriculum for secondary education.

The main objectives of the ICT course for vocational and technical education stated by MoNE (2018c) are professional development, basics of information technologies and programming, technical drawing and office programs, and Computer technical service; Network operation and cybersecurity; Database; and Web programming.

As seen from the ICT course's main objectives for K-12 education, all of these objectives are mainly about twenty-first-century skills such as problem-solving, computational thinking, and creative thinking. On the one hand, the Education Vision of 2023 aims to gain these skills and human beings with various material and spiritual elements.

Computational Thinking

In the twenty-first century, computational thinking is a vital skill that each world citizen should have. Wing (2006), who is a pioneer in the field of computing, defines the computational thinking term as "solving problems, designing systems and understanding of human behavior, using the fundamental concepts of computer science." (p.33).

Furthermore, Wing (2006) also identifies some characteristics of computational thinking. One of the characteristics is conceptualizing, which requires abstract thinking. Another one is about having fundamental skills rather than rote skills. Moreover, it is about how humans think in case of solving problems and mathematical thinking. Another essential characteristic of computational thinking is “ideas,” which Wing (2006) defines as computational concepts that allow us to manage our daily lives using thinking skills. These characteristics suggest that CS education includes principles of computational thinking (Vegas & Fowler, 2020). Accordingly, computational thinking is associated with coding or computer programming. There are several ways to teach children computer programming, such as game-based learning. Based on the National Purchase Diary Panel (NPD) Group report, 73% of children ages 2 and older play video games in the United States (N.P.D Group, 2019). That is, game-based learning could be an effective teaching method for students' learning and motivation (Tatar & Eseryel, 2019).

Programming

It is stated that teaching programming is an essential tool for improving cognitive skills by Nickerson (1983). One of the reasons why programming is crucial is its necessity in people's lives. Nickerson (1983) also states that individuals need programming skills from school to job requirements. Today, many new occupations are being created, and many existing ones are being adapted with new definitions. That’s why, coding becomes a critical talent for those looking for better job opportunities or improving their current skills as an added advantage (Jain, 2021).

On the other hand, programming is generally discussed as demanding and challenging to learn for all ages. However, there are some new initiatives to break down prejudices. For instance, many educational programming tools assist in learning programming. These vary in types, cost, complexity, and learning approach. In this regard, visual programming environments can be an example of one of the educational programming tools (Rose et al., 2020).

Programming Languages

According to Price and Barnes (2015), programming languages can be divided into two types as text-based and visual-based. Traditional or text-based programming languages represent the computers' way of thinking, whereas visual programming languages represent the human language. Therefore, visual programming languages are preferable to text-based for K-12 context (Lye & Koh, 2014). In other words, block-based coding tools have some standard features like having no syntax, drag and drop interface, color and shape, and simplified language

(Moors et al., 2018). Moors et al. (2018) also believe that these features make visual programming easy.

Visual Programming Languages

Before explaining the base of Visual Programming Languages (VPL), it is necessary to define what it is. VPL is defined by Noone and Mooney (2018) as "any programming language where users are able to manipulate the underlying code in some graphical fashion rather than the traditional text-based approach" (p.158).

There has been an educational visual programming language called "Logo," designed in the 1960s by Seymour Papert. He created it to develop students' thinking and solve problems regarding teaching computer programming. According to the review of Papert's dream by Resnick (2012), computers' costs still were too much, and personal computers were not produced yet in the past, meaning it was hard to reach computers. However, he still believed that everyone, including children, would get the computers one day and establish informatics basics regarding children's way of thinking and playing, which comes true. Even younger kids have been interacting with computing somehow by electronic devices for communicating, searching, or playing games today. However, from Papert's view, children should be interacting with these technologies and program their projects. Therefore, numerous students used Papert's Logo programming language when personal computers became available, but they had difficulty because of its syntax. Later on, the developed form of the Logo programming language was designed called "Scratch," which makes programming more thinkable, meaningful, and social. Thus, the visual programming language has begun.

Block-Based Coding as a Visual Programming Language

Block-based coding, also known as block-based programming, is a form of visual programming language in which instructions or commands are fundamentally depicted as blocks by dragging and dropping them (“Block-Based Coding,” 2020). In other words, it is like an educational programming environment in which the codes are performed graphically rather than text. Noone and Mooney (2018) indicated that the drag-and-drop approach fosters creativity in learning which might not be possible with text-based coding. Resnick et al. (2009) also stated that these environments offer ease of use, chances of making complicated and creative projects through time and support various projects. Furthermore, block-based coding environments provide learners with a chance to experiment more and easier when they see the outcomes of what they created (Noone & Mooney, 2018). What is more, Kalelioglu et al. (2014) claimed that block-based coding environments promote the teaching of coding to students without much memorization because the curriculum expects students to improve their problem-solving and computational thinking skills.

Common Block-Based Coding Environments at K-12 Level

Lately, there are plenty of widely used block-based programming environments have emerged. Some of them are paid, and some of them are unpaid to use. Common Sense Education, a non-profit organization, provides the trustworthy information to improve learners' skills in the twenty-first century (Common Sense, n.d.). In this context, this organization has listed the programming languages for each stage as elementary, middle, and high school, respectively. Table 1 shows the common ones that are used at K-12 level.

Table 1

List and a Short Description of the Well-known Block-based Coding Environments at K-12 Level (Common Sense Education, 2020a; Common Sense Education, 2020b; Common Sense Education, 2020c)

Block-based Coding Environment	Short Description	Available in
Scratch	It is a non-profit programming language that enables to create interactive stories, games, and animations.	https://scratch.mit.edu/
Code.org	It aims to encourage students to learn computer science by free coding lessons grade by grade.	https://code.org/
Tynker	It is a visual coding tool that teaches kids to program with blocks of code.	https://www.tynker.com/
Alice	It is a 3D-oriented programming language to create animations through a drag and drop graphical interface.	https://www.alice.org/
KoduGame	It is a 3D game-making environment that has visual and straightforward logic.	https://www.kodugamelab.com/
Kodable	It introduces the fundamental concepts of computer programming through a series of games	https://www.kodable.com/

Conclusion of the Literature Review

All in all, having twenty-first-century skills is significant for students' lives in today’s world. They should have these skills regarding analyzing and solving problems critically and thinking creatively. According to the literature review, some of the twenty-first-century skills can be developed through block-based coding environments. Furthermore, computational thinking has an important place in coding skills. Although the visual programming language's history goes back a long way, block-based coding is a relatively trending topic nowadays. Besides this, coding is one of the fundamentals of the curriculum.

Considering the learning theories, approaches, and ideologies, computer science education is based on the constructivist theory, mostly project and problem-based approaches. Moreover, it can be associated with both learner-centered and social-efficiency ideology and post-modern curriculum as well. In addition to these, there is a need to provide equity in terms of gender diversity in CS education and all educational fields. Therefore, there could be some ways to provide equitable pathways for computer science learning and encourage heterogeneity in computer science classrooms by applying for mentorship programs, redefining the field, and providing opportunities for engagement like competitions and camps.

In conclusion, this study helps identify the year of the publications, research methodologies, data collection instruments, the grade level of the sample, sampling sizes, countries, subject areas, block-based coding tools, and teaching/learning approaches comprehensively and holistically. This study also benefits to find out which learning domains were explored and results of those studies regarding the impact of block-based coding in K-12 education on learning domains in the research articles. Therefore, there is a need to conduct a systematic literature review in this context so as to be able to identify the aforementioned subjects in detail with further analysis.

CHAPTER 3: METHOD

Introduction

In this chapter, how the methodological procedures of the present study are utilized is explained. Initially, the description of the research design is presented comprehensively. Secondly, the sampling method is explained. Thirdly, clarification of the process of instrumentation is done. Followingly, explaining the methodology of the current study is finalized by providing information about the method of data collection and data analysis of the study. Consequently, the credibility of the study is explained under two headings, namely reliability and validity.

Research Design

This is a qualitative research study, and the thematic content analysis method, which can also be named as systematic literature review, was used (Caulfield, 2021). According to the Çalık and Sözbilir (2014), thematic content analysis synthesizes and criticizes the results of research articles conducted in a specific content area by creating themes. Moreover, content analysis is defined by Krippendorf (2004) as “analysis of the manifest and latent content of a body of communicated material (as a book or film) through classification, tabulation, and evaluation of its key symbols and themes to ascertain its meaning and probable effect” (p. xvii). On another note, Braun and Clarke (2006) also defined thematic content analysis as “a method for identifying, analyzing and reporting patterns or themes within data” (p.79).

In this regard, the thematic content analysis method was used to identify demographic characteristics of research articles and investigate which learning domains were explored in the research articles. Besides, results were revealed by themes regarding the impact of block-based coding in K-12 education on learning domains.

Sample

This study's sample is composed of scholarly articles that are written in K-12 education about block-based coding. Initially, keywords are determined to identify the target population. The keywords used for the query included “block-based coding,” “block-based programming,” “visual programming language,” “education,” “K-12 education,” “school,” “preschool,” “kindergarten,” “primary education,” “secondary education,” “high school,” “middle school,” “scratch,” “code org,” “alice,” and “computational thinking.” What is more, database search engines that are Web of Science and Scopus were reviewed by determining keywords. The time span for the study was limited to be from 2000 to 2020. Also, results were restricted to only the English language. Furthermore, from the document types, only “articles” were selected to refine the criteria.

In this study, there are criteria to exclude some research articles. One of them is that consider other than K-12 level and has the sample of the studies which consider teachers and pre-service teachers. In addition, the studies which consider gaming in second life and physical devices supporting the learning of programming such as tangible construction kits and educational robotics and applications were excluded. After that, the studies which do not match with keywords and have the studies which have book chapters, conference proceedings, and idea-based papers were eliminated.

Instrumentation

Each selected research article was categorized and labelled in this study using a codebook on Spreadsheet Software. The researcher developed a codebook.

Neuendorf (2017) stated that coders use electronic coding forms (e.g., Microsoft Excel) instead of traditional papers because of their easiness and quickness with technology advances.

The categories include these headings respectively: the demographic characteristics of the research articles in the literature about block-based coding in K-12 education, including years of the publications, research methodologies, data collection instruments, the grade level of the sample of the study, sampling sizes, countries researches carried out, subject areas covered in the articles, block-based coding tools, teaching/learning approaches.

Besides the demographic characteristics of the research articles in the literature about block-based coding in K-12 education, learning domains explored in the research articles were written on the Spreadsheet Software. Finally, results of the research articles were noted down regarding the impacts of block-based coding in K-12 education on learning domains.

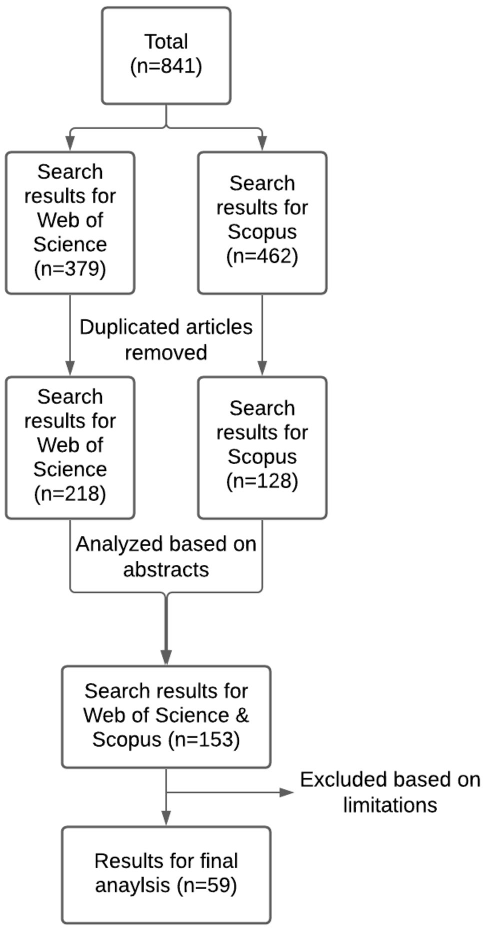
Method of Data Collection

A total of 59 research articles were listed by determining keywords. The articles to be examined within the scope of the research were obtained by scanning Web of Science and Scopus databases. Besides, they were limited to the years between 2000 and 2020. As a result of the preliminary examination of the articles that are appropriate for the purpose and contain necessary information was accessed.

All of the required restrictions and exclusion criteria were entered as stated in the sampling part. In the beginning, the study made use of the Web of Science and Scopus databases and only research articles reviewed by entering determined keywords. After reaching the search results for Web of Science and Scopus, 841 research articles were reached on the 8th of October in 2020 (See Figure1). The researcher refined research adding new keywords on the 11th of October, 2020. Out of the 841 research articles, duplicated articles were removed. One article was added on the first of December in 2020 since the researcher received a notification about the new article being published on the same keywords. After that, the rest of the research articles were analyzed based on abstracts, and redundant ones were also eliminated. In the end, the sampled research articles from both databases were combined. In the last phase, the ones outside the scope of K-12 education, which consider teachers and pre-service teachers, and which consider gaming in second life and physical devices supporting the learning of programming such as tangible construction kits and educational robotics and applications, were left aside.

Figure 1

Research Articles Selection Process



Methods of Data Analysis

In this study, the research articles were analyzed comprehensively and holistically. Braun and Clarke (2006) identified six phases for conducting thematic analysis. Thus, these phases were considered for data analysis in this study. Initially, the first phase is about familiarizing yourself with your data. It is about reading, transcribing, and noting down the initial ideas of the data. It means the data, which are research articles, were read comprehensively and took some notes by highlighting while reading. The second phase is about generating initial codes. That is, the features of data were coded systematically. The third phase is about searching for themes. It means that the generated codes were blended into specific themes. The fourth phase is about reviewing the themes. That is to say, the themes were checked whether they were correlated with the coded extracts and the entire data set. The fifth phase is about defining and naming themes. In this phase, clear definitions and names were generated for each theme. The last and sixth phase is about producing the report. To clarify, a scholarly report of the final analysis of research articles was produced.

Validity and Reliability

According to Creswell (2014), validity in findings means that the researcher determines the accuracy or credibility of the results through strategies such as member checking. Therefore, inter-coder agreement (or cross-checking) was used in this study (Creswell, 2009). The research articles selected for the reliability of the research were examined in detail by another researcher who is a master's student and interested in the educational technology field. The researcher gave the data to cross-check in January 2021. The results were received one week later confirming the validity data. According to Miles and Huberman (1994), reliability is calculated by using this formula: Reliability = number of agreements / (number of agreements + number of disagreements). Thus, 10% of the research articles were selected randomly and given to the inter-coder. The inter-coder was instructed to use the codebook to analyze the research articles. The results showed that 80% agreement was achieved. After that, the researcher and the other evaluator shared ideas after the coding process and reached common decisions. In addition, the supervisor of this study, who is an expert in this field and has published many qualitative studies, also controlled all the processes. In the end, all controls and feedbacks of the supervisor have been checked by the researcher and finalized.

CHAPTER 4: RESULTS

Introduction

This chapter gives detailed information about the study's findings obtained from a thematic content analysis of the research articles related to block-based coding in K-12 education. The results of each research question are presented using percentages and frequencies in tables and figures.

Findings of the Study

This study presented the findings of 59 studies conducted within the framework of the use of block-based coding in K-12 education published from 2000 to 2020 (See Appendix A).

The Demographic Characteristics of the Research Articles in the Literature about Block-based Coding in K-12 education

Year of the Publications

Table 2 shows the distribution of the research articles regarding block-based coding in K-12 education according to the years by publication time.

Table 2

Distribution of Research Articles by Years

Years	f
2020	12
2019	14
2018	8
2017	9
2016	3

Table 2 (cont'd)

Distribution of Research Articles by Years

Years	f
2015	7
2014	3
2013	2
2010	1

When Table 2 was examined, it was found that an increase was observed in research articles published in 2019 (*n*=14). There were also notable number of studies in 2020. Besides, there was also a considerable number of studies in 2018 (*n*=8) and 2017 (*n*=9). However, there was no study published before 2010 and between 2010 and 2013.

Research Methodologies

The findings of the research methods used in the research articles are classified as; quantitative, qualitative, and mixed and are given in Table 3.

Table 3

Distribution of Research Articles by Research Methodologies

Methods	f
Mixed	29
Quantitative	19
Qualitative	8
Unspecified	3

When Table 3 was examined, it was seen that mixed (*n*=29) and quantitative (*n*=19) methods were primarily used, whereas the qualitative method (*n*=8) was not relatively preferred. However, research methodologies of some of the research articles were not specified (*n*=3).

Data Collection Instruments

The distribution of data collection instruments used in the research articles is presented in Table 4.

Table 4

Distribution of Instruments of the Research Articles

Instruments	f
Assessments	35
Questionnaires	25
Interviews	18
Assignments	18
Observations	16
Others	8
Scales	7
Diaries	2
Rubrics	3
Screen records/screenshots	2
Eye-tracking measures	2
Instructor reflections	1

The analysis showed that the most preferred data collections tools are assessments (*n*=35). Assessments are used to test skills or knowledge to see the progress, and include Visual Blocks Creative Computing Test (VBCCT), Computational Thinking Tests, Abstraction Tests, Academic Achievement Tests, Knowledge Acquisition Tests such as Programming Knowledge, and Skill Tests, and Motivations Tests.

Furthermore, questionnaires (*n*=25), interviews (*n*=18), assignments (*n*=18), and observations (*n*=16) are the frequently preferred data collection tools. Questionnaires include students' written responses to reflective statement questionnaires, attitudinal questionnaires, critical thinking questionnaires, and surveys. Surveys consist of attitudinal surveys such as computer science attitude surveys. Besides, interview types are semi-structured and post-hoc interviews. Additionally, assignments are also used relatively. Assignments include projects (e.g., students' portfolios, games, and weekly interim and final Scratch programs), tasks, challenges, worksheets, activities, and artifacts constructed by the children. Moreover, observations include participant observations, field notes, classroom observations, observational research, and learning observation documents.

In addition, scales (*n*=7) and others (*n*=8) were also used in research articles moderately. Scales involve cognitive load scale, problem-solving skills scale, self-efficacy perception scale, and computer-based skill level scale. Additionally, others include background information of students, and interventions. Lastly, as it is seen in Table 4, diaries, rubrics, screen records or screenshots, eye-tracking measures, and instructor reflections were used slightly.

The Grade Level of the Sample of the Study

The grade level of the sample of research articles examined were given in Table 5.

Table 5

Distribution of Grade Level of the Sample of Research Articles

The Grade Level	f
Middle school students	37
Elementary school students	13
High school students	9
Pre-school students	1

When Table 5 was investigated, it was found that most of the research articles were conducted with middle school students (*n*=37). There were also research articles targeting elementary school students (*n*=13) and high school students reasonably (*n*=9). On the contrary, the least number of research articles was observed with pre-school students (*n*=1).

Sampling Sizes

The sampling size of the research articles examined is reported in Table 6.

Table 6

Distribution of Sample Sizes of the Research Articles

Sample Size	f
Less than 50	23
Over 100	20
Between 50 to 100	14
Unspecified	2

Concerning the sample size of the research articles, a significant proportion of the studies were with less than 50 sample sizes (*n*=23) as shown at Table 6. The next most frequently used sample size was over 100 (*n*=20). Very few of the studies sampled ranging from 50 to 100 (*n*=14). Lastly, sample size of some of the research articles were not specified (*n*=2).

Countries Researches Carried Out

The distribution of countries researches carried out is presented in Table 7.

Table 7

Distribution of Countries of the Research Articles

Countries	f
United States	12
Spain	11
Turkey	6
Greece	3
Croatia	3
China	2
England	3
Israel	2
Korea	2
Norway	2
Scotland	2
Serbia	2
Taiwan	2
Columbia	1
Cyprus	1
Czech Republic	1
Georgia	1

When Table 7 is analyzed, it was observed that research articles were conducted mainly in the United States (*n*=12), Spain (*n*=11), and Turkey (*n*=6). Then, Greece (*n*=3) and Croatia (*n*=3) followed. However, the rest of the countries have a few studies in which carried out.

Subject Areas Covered in the Research Articles

Within the scope of the research, the disciplines (subject areas) in which block-based coding was used were determined to reveal the potential of block-based coding in the educational field and to evaluate the results. The disciplines were separated into six groups: Computer Science, Science (including Chemistry and Physics), Mathematics, Music, Arts, and Social Sciences.

Table 8

Subject Areas Covered in the Research Articles

Subject Areas	f
Computer Science	45
Mathematics	7
Science	5
Music	2
Art	1
Social Sciences	1

Block-based coding was primarily used in Computer Science lessons with 45 research articles considering Table 8. Moreover, Mathematics (*n*=7) and Science (*n*=5) fields were also used, respectively. However, not many research articles were found in the literature in terms of the use of the areas of Music (*n*=2), Arts (*n*=1), and Social Sciences (*n*=1).

Block-based Coding Tools

In this study, block-based coding tools used in the research articles is also examined in Table 9.

Table 9

Block-based Coding Tools of the Research Articles

Block-based Coding Tools	f
Scratch	36
Code.org	10
Alice	4
App Inventor	2
Scratch4OS	2
Pencil Code	2
ScratchJr	2
mBot	1
Micro:bit	1
Netsblox	1
Penguin Go	1
RoBlock	1
CompThink	1
Logo	1
Python	1
Pirate Plunder	1
Lightbot	1

Table 10 (cont'd)

Block-based Coding Tools of the Research Articles

Block-based Coding Tools	f
Turtle Blocks JS	1

Considering the Table 9, it is seen that there was a tendency toward Scratch (*n*=36) among block-based coding tools. On the other hand, it is seen that Code.org (*n*=10) was used relatively underutilized. Additionally, Alice is used very few compared to Scratch and Code.org.

Teaching/Learning Approaches

The teaching/learning approaches used in the research articles examined are listed in Table 10.

Table 11

Teaching/Learning Approaches used in the Research Articles

Teaching/Learning Approaches	f
Problem-based learning	24
Project-based learning	16
Interdisciplinary learning	15
Game-based learning	11
Collaborative-learning	3

As given in Table 10, the problem-based learning approach was the most used among the teaching/learning approaches. Project-based learning and interdisciplinary learning approaches were found to be close to each other. Besides, game-based learning approach was used relatively less than the aforementioned approaches. The least used approach was collaborative learning in the research articles.

Learning Domains Explored in the Research Articles

The learning domains explored in the research articles about block-based coding in K-12 are reported under three categories: cognitive, affective, and both cognitive and affective domains.

Table 12

Distribution of Research Articles Regarding Learning Domains

Learning Domains	f
Cognitive	34
Cognitive & Affective	21
Affective	4

When Table 11 was investigated, it appears that cognitive domain (*n*=34) was studied mostly in the research articles. Following this, both cognitive and affective domain (*n*=21) were explored together in the research articles respectively. However, a few of the research articles studied affective domain (*n*=4).

The Impact of Learning Domains on Block-based Coding in K-12 Education

In this section, the results of research articles are analyzed regarding the impact of learning domains on block-based coding in K-12 education. That is, the themes of the research articles were examined, and they were divided into major themes and each theme was examined with their sub-themes in consideration.

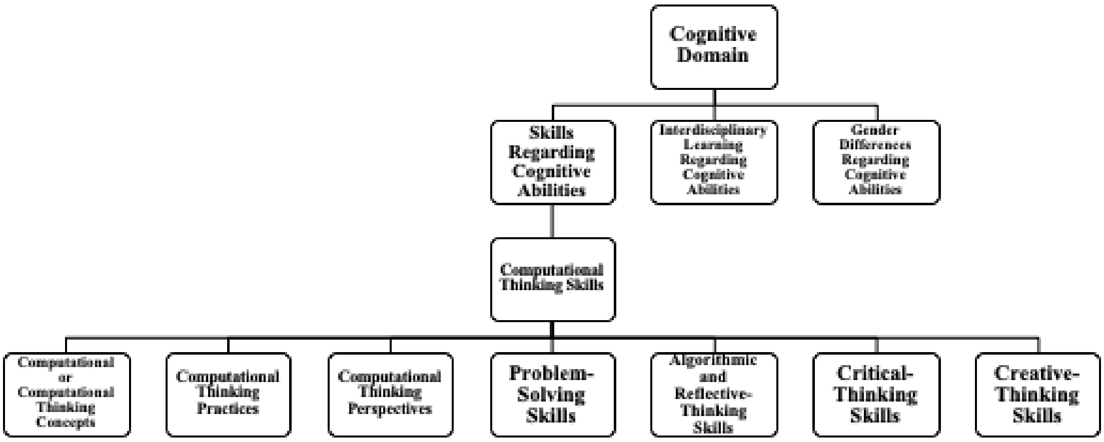
Cognitive Domain

Three major themes were found in terms of the cognitive domain: Skills Regarding Cognitive Abilities, Interdisciplinary Learning Regarding Cognitive Abilities, Gender Differences Regarding Cognitive Abilities (See Figure 2).

One of the themes is found as skills regarding cognitive abilities, and computational thinking skills is the initial one. Computational thinking skills also have sub-themes include “Computational or Computational Thinking Concepts; Computational Thinking Practices; Computational Thinking Perspectives” (Deng et al., 2019, p.91); Problem-Solving Skills; Algorithmic and Reflective-Thinking Skills; Critical-Thinking Skills; and Creative-Thinking Skills. Under the Computational Thinking Concepts, there are “sequences, loops, events, parallels, and conditionals” (Deng et al, 2019, p.91). As for the Computational Thinking Practices, they are “testing, debugging, reusing, and remixing” (Deng et al., 2019, p.91). Lastly, Computational Thinking Perspectives include “expressing, connecting, and questioning” (Deng et al., 2019, p.21). Following this, the second major theme is found as Interdisciplinary Learning Regarding Cognitive Abilities. Finally, the third major theme is found as Gender Differences Regarding Cognitive Abilities.

Figure 2

Themes Regarding Cognitive Domain



Skills Regarding Cognitive Abilities

Block-based coding environments affect the process of learning in terms of many aspects, such as computational thinking skills. The findings of research articles about computational-thinking skills are given in this section regarding both improvements and opposing sides.

Computational Thinking Skills

Initially, Aksit and Wiebe (2019); Broll et al. (2018); Çakıroğlu and Mumcu (2020); del Olmo-Muñoz et al. (2020); Moreno-León et al. (2015); Rose et al. (2020); Pérez-Marín et al. (2020); Sáez-López and Sevillano-García (2017) reported that students' computational thinking skills are developed with coding in block-based coding environments. Moreover, it was written by Lawanto et al. (2017) that there is an improvement in some of the computational thinking skills, which are synchronization, parallelism, and flow control. Further in line, five articles found that block-based coding environments foster computational thinking competencies and improve "computational thinking concepts, computational thinking practices, and computational thinking perspectives" (Chou, 2019; Deng et al., 2019; Sáez-López et al., 2016, 2019; Tran, 2018). Çakıroğlu et al. (2018) also concluded that "the highest cognitive load level was recorded in the sequences concept" (p.16) such as repeat, forever, and repeat until blocks (Lawanto et al., 2017), and "the lowest perceived cognitive load was observed in the loop concept such as nested loops" (Çakıroğlu et al., 2018, p.16).

Besides, researches by Çakıroğlu et al. (2018); Kalelioğlu (2015); and Rose et al. (2020) found that block-based coding environments provide students with many advantages in terms of learning basics of computer science and understanding basic computational concepts with the help of "motivating features, helpful instructional materials, and lesson plans inside" (Kalelioğlu, 2015, p.209). Wang et al. (2017) also stated that students showed better programming knowledge and skills using block-based coding environments.

On the one hand, some of the research articles compared text-based and block-based coding environments regarding the learning of computational thinking concepts. According to these comparisons, Mladenović et al. (2020) concluded that students got above scores when using block-based programming than text-based. Furthermore, "students' misconceptions about programming concepts, loops, are minimized when using Scratch, a block-based programming language, compared to text-based programming languages, which are Logo and Python" (Mladenović et al., 2017, p.1498). Moreover, Papadakis and Orfanakis (2018) found that AppInventor seems to emerge as a more powerful novice programming environment for adoption in secondary education compared to Alice in terms of learning basic programming concepts. In addition to these, Wilson et al. (2013) found that "the most used programming concepts used by children were User Interaction (key handling), Loops (iteration), and Conditional Statements. Also, random numbers were the lowest utilized programming concept" (p.106).

Furthermore, Estevez et al. (2019) explored the impact of AI (Artificial Intelligence) in everyday life. They concluded that students understood the underlying computational thinking skills of the AI process by using Scratch. Besides, students who had learnt Scratch had a lot of programming knowledge and expertise, which made studying the more complex subjects a lot easier. (Armoni et al., 2015).

In contrary to previous studies, some studies found opposing sides of computational thinking skills. Lawanto et al. (2017) stated a weakness in computational thinking skills that are data representation and abstraction skills of students. Additionally, Chou (2019) found no improvement in one of the computational thinking concepts, loops. Moreover, Çakıroğlu et al. (2018) concluded that it's difficult to use Scratch's interface for activities that require deeper programming ideas such as one of the programming concepts, nested. What is more, Chang (2019) investigated the interrelationship between the learning of computational thinking concepts and the practice of digital curation through digital storytelling. Thus, it was found that they had difficulty to incorporate more abstract and complicated computational thinking concepts into digital storytelling tasks. Moreover, Meerbaum-Salant et al. (2013) reported problems in certain topics to be taught such as "loops (repeated execution), variables, and concurrency" (p.261).

Problem-Solving Skills

In accordance with the findings of the study, researches by Çakıroğlu et al. (2018) and Tran (2018) found that block-based coding environments increase students' problem-solving skills. Also, problem-solving steps are completed by students in a sequential manner (Çakıroğlu & Mumcu, 2020). In addition, Tangney et al. (2010) also found that students were deeply engaged in difficult problem-solving and programming activities. Denner et al. (2014) also found that students work on complicated problem-solving during programming.

However, Çiftci and Bildiren (2019) reported no effect on problem-solving skills. Also, App Inventor did not improve the elementary school students' problem-solving capabilities (Kim & Lee, 2017). While compared to text-based programming languages, students with poorer problem-solving ability are more successful when programming with Scratch (Mladenović et al., 2017).

Algorithmic and Reflective Thinking Skills

Other findings regarding Algorithmic and Reflective Thinking Skills, researches by Burker (2015); Feijoo Garcia and De la Rosa (2016); Gruenbaum (2014); Hainey et al. (2019); Meerbaum-Salant et al. (2013); Mladenović et al. (2017); Moreno-León et.al (2015); Pérez-Marín et al. (2020); and Seralidou and Douligeris (2020) reported that using block-based coding can facilitate the process of learning algorithmic thinking concepts and skills. Additionally, Erümit (2020) also revealed that the mathematical and game preparation activities positively affected algorithmic thinking and reflective thinking skills for problem-solving.

Critical Thinking Skills

Another result was revealed by the findings of the current content analysis indicating that students gain a deeper understanding of critical thinking in the Scratch programming environment with the peer-assessment technique (Wang et al., 2017).

Creative Thinking Skills

What is more, Broll et al. (2018); Burker (2015); and Hainey et al. (2019) revealed that students could improve their creative thinking skills with the help of block-based coding environments. It means these environments encourage students to be creators (Broll et al., 2018). For instance, block-based coding environments provide building procedures such as 3D printings (Burker, 2015). Furthermore, Scratch is an effective and applicable tool for creative thinking development in primary education (Hainey et al., 2019). Additionally, Iskrenovic-Momcilovic (2019) revealed that visual programming language environments make learning more creative.

Interdisciplinary Learning Regarding Cognitive Abilities

The second major theme is Interdisciplinary Learning Regarding Cognitive Abilities. It was found that block-based coding environments were integrated into six curricular areas: Mathematics, Social Sciences, Science, Music, and Art. In this respect, Akpinar and Aslan (2015); Ke (2014), Sáez-López et al. (2019); Ortiz-Colon and Maroto Romo (2016); Vidal-Silva et al. (2020) reported that benefits were acquired in the understanding of mathematical concepts, thinking logically and mathematically, and developing algorithmic and logical-mathematical competencies within the integration of programming into the math curriculum. In addition, Kalelioğlu (2015) reported that using the Code.org site, which is a block-based coding tool, teaches programming and improves mathematical and geometrical knowledge. Also, the students used computational abstractions in the form of variables and used various ways to create mathematical models in a programming context (Ng & Cui, 2020). However, Jordonova et al. (2019) stated that the level of mathematical skills requirements in activities is noticeably lower than the level of programming requirements. It has proven appropriate to modify mathematics content to a higher level, or simplify the programming tasks. This will improve students' mathematical skills by creating new and better tasks.

Further on the research articles, Aksit and Wiebe (2019) and Puttick and Tucker-Raymond (2018) showed that designing games in block-based coding environments foster students' understanding of science concepts. However, Sáez-López et al. (2019) reported that there is not an improvement in the science subject area. Lastly, two research articles reported that students learn and understand the many facts of the chemistry concepts while experiencing in the visual programming environment (Loganathan et al., 2019; Kim, Choi, & Paik, 2019).

Furthermore, Sáez-López et al. (2016) reported an improvement in content creation in art and history with an active pedagogical approach in a block-based coding environment. Moreover, students successfully use the initialization and synchronization strategies, the programming concepts, to produce cohesive music videos using Scratch (Fields et al., 2015). In addition, students can create their instruments and program them to make Music so that their creativity is developed within a technological and artistic sphere (Sáez-López & Sevillano-García, 2017).

Gender Differences Regarding Cognitive Abilities

The third major theme is Gender Differences Regarding Cognitive Abilities. Within the scope of this theme, Puttick and Tucker-Raymond (2018) revealed that designing games or programming at block-based coding environments positively affected females' learning. Female students were just as effective as male students, even if they said they had never heard of coding or were terrified of it. Also, female students even noted that programming could be a viable option for them in the future (Kalelioğlu, 2015; Papavlasopoulou et al., 2020).

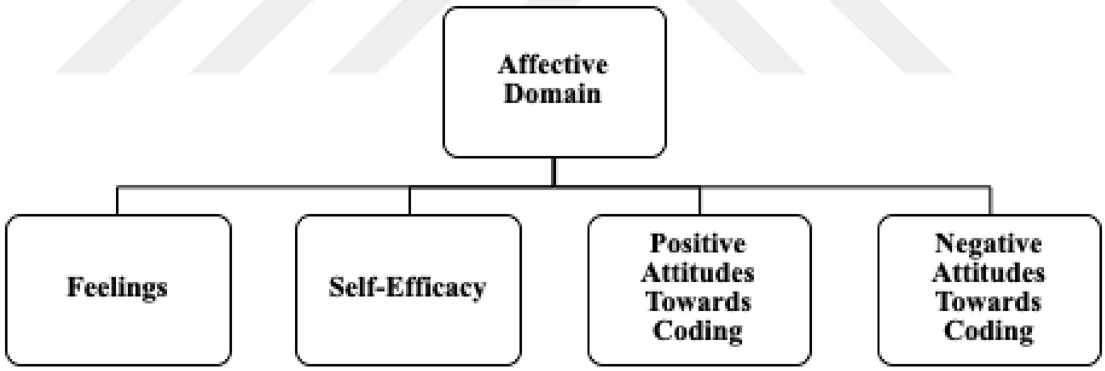
On the contrary, there were no gender disparities in primary school students' reflective thinking skills toward problem-solving while they programmed on the Code.org site. Female students' reflective thinking abilities for problem-solving are slightly better than male students' reflective thinking skills for problem-solving (Kalelioğlu, 2015). On the one hand, Papavlasopoulou et al. (2020) also found no gender differences in students' cognitive processes regarding the learning of programming.

Affective Domain

According to the research articles regarding the affective domain, there are both improvements and declines in students’ skills. Four different themes were found in terms of the affective domain (See Figure 3). The themes are; Feelings; Self-Efficacy; Positive Attitudes Towards Coding; Negative Attitudes Towards Coding.

Figure 3

Themes Regarding Affective Domain



Feelings

The feelings include “emotions, interactions, and how students deal with things emotionally like values, appreciation, enthusiasm, confidence, and motivation” (Patson, 2020, para.4). In this regard, Armoni et.al (2015); Broll et al. (2018); Chou (2019); Deng et al. (2019); Mladenović et.al (2017); Papavlasopoulou et al. (2019); Papavlasopoulou et al. (2020); Papadakis and Orfanakis (2018); Sáez-

López et al. (2016; 2019); Sáez-López and Sevillano-García (2017); and Theodoropoulos et al. (2017) reported that there is an improvement in motivation, engagement, and commitment to code due to block-based coding environments. Also, block-based coding environments provide students with fun (enjoyment), sociability, involvement, cooperation with other peers, participation, collaboration, enthusiasm, curiosity, and interest in the subject matter and future computing courses. It was also stated that programming in pairs results in increased motivation for learning new skills, faster recall of new concepts, and more effective learning acquisition of programming languages (Iskrenovic-Momcilovic, 2019; Weintrop & Wilensky, 2017). Similarly, Iskrenovic-Momcilovic (2019) also found that block-based coding environments allow learning to become more exciting and more challenging. In addition to this, these environments are also nice, engaging, and friendly tools (Feijoo Garcia & De la Rosa, 2016).

Moreover, Gruenbaum (2014) found that four times as many boys as girls signed up to participate in coding activities, despite considerable initial interest from girls. Moreover, del Olmo-Muñoz et al. (2020) found a gap in which girls were less motivated than boys.

However, while Basogain et al. (2018) reported that block-based coding environments increase confidence in students' skills and knowledge in programming, Weintrop and Wilensky (2017) said that there was no difference between students in the two conditions that are block-based and text-based in terms of confidence and enjoyment.

Self-Efficacy in Coding

According to the Soykan and Kanbul (2018), students who had coding education demonstrated stronger self-efficacy in programming than students who had not. On the contrary, App Inventor in the educational program did not improve the elementary school students' self-efficacy because they needed more class time due to the difficulty of learning blocks in the App Inventor educational program (Kim & Lee, 2017). Furthermore, there was a gender difference found in students' perceptions which means girls have less self-efficacy in programming than boys (Papavlasopoulou et al., 2020).

Positive Attitudes Towards Coding

Nine research articles have been analyzed that suggests using block-based coding tools in K-12 education improves students' attitudes towards computer science education. Deng et al. (2019), Kalelioğlu (2015), Papavlasopoulou et al. (2019), Ortiz-Colon and Maroto Romo (2016), Tangney et al. (2010), and Wang et al. (2017) reported that there is an improvement in students' attitudes regarding learning programming. It means students are likely to show positive attitudes toward the programming course.

Besides, there were also comparisons between block-based and text-based coding tools regarding positive attitudes. Mladenović et al. (2017) found that students have positive attitudes towards Scratch than Python. According to Maloney et al. (2010), Scratch makes coding enjoyable and visual by creating animated stories and games. Likewise, Erümit (2020) concluded that animation activities positively affected students' attitudes towards computer technologies.

Negative Attitudes Towards Coding

Two research articles reported that there are negative changes in students' attitudes toward computer science. This is because of the challenges of additional constraints, which is a specific game feature (Zhao & Shute, 2019) and lack of algorithmic thinking and problem-solving skills in younger students (Lambić et al., 2020).

CHAPTER 5: DISCUSSION

Introduction

In this chapter, initially, an overview of the study is given. Secondly, the major findings are discussed. Following this, implications for practice and further research are stated.

Overview of the Study

In this study, research articles about block-based coding in K-12 education published from 2000 to 2020 were examined, and 59 research articles that met the criteria were analyzed through Web of Science and Scopus databases. Thematic content analysis was used as a research methodology to explore demographic characteristics of the research articles, including years, research methodologies, data collection instruments, the grade level of samples and their sizes, countries in which researches were carried out, subject areas covered, block-based coding tools and teaching/learning approaches used in the articles. These selected articles were investigated by using a codebook in a Spreadsheet software. Besides, which learning domains were explored in the research articles and the results of those studies regarding the impact of block-based coding in K-12 education on learning domains were investigated. The following section discusses the major findings and possible reasons for these findings under subtitles.

Discussion of Major Findings

The Demographic Characteristics of the Research Articles about Block-based Coding in K-12 Education

According to the distribution of research articles about block-based coding in K-12 education across years, it seems that there is a sharp increase in the number of research articles from 2000 to 2020. Most of the research articles were published in the last four years, especially in 2019. There were fewer research articles published in 2020 compared to 2019. The reason for the decrease in this year might be because of pandemic, during which researchers could not conduct as many studies. It is also seen that there were no articles published before 2010, which might be because of the fact that coding received a place in the ICT curriculum and gained popularity after 2010.

In accordance with the findings of the study, the most preferred research method was found as mixed method. Considering that the mixed research method was used in the research articles, it is concluded that the most used data collection tools are achievement tests (See Table 4).

As this study has shown, the research articles were generally carried out at the middle school level. This is because block-based coding environments can be more suitable for middle school levels than other levels considering the fact that a well-developed ICT curriculum, would support middle school students' future career decisions via improving their CT skills (Rich & Hodges, 2017). Also, most of the studies were conducted with less than 50 sample sizes. The underlying reason is that conducting studies with small groups can be more efficient than larger ones.

This review also found that most research studies were conducted in the United States, even though England first attempted to integrate coding into the curriculum (European Schoolnet, 2015). The United States can give weight to educational technology more since as many as 47 percent of future jobs will become automated entirely in the United States in the next 20 years (Polyviou, 2021). On the other hand, it drew attention that Turkey has very few studies conducted about block-based coding in K-12 education (Akpinar & Aslan, 2015; Çakıroğlu et al., 2018; Çakıroğlu & Mumcu, 2020; Çiftçi & Bildiren, 2020; Erümit, 2020; Kalelioğlu, 2015).

When the obtained results were analyzed, it was understood that the studies were generally in the Computer Science field. This was an expected result since the research articles' main topic is generally about computing. Besides, Mathematics and Science subject areas were also covered in the articles moderately. However, Biology was not included under the Science subject area even though there are many developments regarding computational thinking skills in that subject area. For instance, many Science projects, including Biology projects done in Code.org, regarding an hour of code activities (Code.org, n.d).

Further on research articles, Scratch was the most used one as a block-based coding tool. The underlying reason for this could be its popularity (Resnick et al., 2009). However, some common block-based coding tools have not been encountered like Tynker, Kodu Game, and Kodable despite being well-known tools in education (See Table 1). Although many recent block-based coding tools were found, such as Pencil Code, mBot, Netsblox, Penguin Go, RoBlock, CompThink, Pirate Plunder, and Turtle Blocks JS, the number of research articles that focuses on these tools was very few in the literature.

Considering the learning approaches used in the research articles, the most used was problem-based approach. Also, project-based and interdisciplinary-learning approaches were found to be used often, as well although not as much as problem-based approach. As for the benefits of these approaches, Westwood (2008) stated that these approaches allow students to connect curricular knowledge to real-world situations and apply skills and techniques.

Learning Domains Explored in The Research Articles

Primarily, the cognitive domain was covered when learning domains were explored in the research articles about block-based coding in K-12. However, any study that considers the psychomotor domain was not found. In my opinion, there should have been more research exploring the psychomotor domain to see the development of students' skills, such as drag-and-drop skills, by using block-based coding tools. The reason for this could be that young students' psychomotor skills have not fully matured, so they are unfamiliar with computer items such as the keyboard and mouse (Reddy, 2021).

The Impact of Learning Domains on Block-Based Coding in K-12 Education

Regarding the results concerning the cognitive domain, findings of the study signify that block-based coding tools improve computational thinking skills in K-12 education (Aksit & Wiebe 2019; Broll et al., 2018; Çakıroğlu & Mumcu, 2020; del Olmo-Muñoz et al., 2020; Lawanto et al., 2017; Moreno-León et al., 2015; Pérez-Marín et al., 2020; Rose et al., 2020; Sáez-López & Sevillano-García, 2017). Besides, there are also improvements in “computational thinking concepts, computational thinking practices, and computational thinking perspectives” (Chou, 2019; Çakıroğlu et al., 2018, Deng et al., 2019; Sáez-López et al., 2016, 2019; Tran, 2018). Notwithstanding the improvements, opposing sides are also found in computational thinking concepts regarding computational thinking skills (Chang, 2019; Chou, 2019; Çakıroğlu et al., 2018; Lawanto et al., 2017; Meerbaum-Salant et al., 2013).

The findings of this study underline that block-based coding increase students' problem-solving skills (Çakıroğlu et al., 2018; Çakıroğlu & Mumcu, 2020; Denner et al., 2014; Tangney et al., 2010; Tran, 2018). Moreover, it can be stated that coding with block-based environments facilitates the process of learning algorithmic and reflective thinking skills, which might be a result of teaching and learning methods employed in class (Burker, 2015; Erümit, 2020; Feijoo Garcia & De la Rosa, 2016; Gruenbaum, 2014; Hainey et al., 2019; Meerbaum-Salant et al., 2013; Mladenović et al., 2017; Moreno-León et al., 2015; Pérez-Marín et al., 2020; Seralidou & Douligeris, 2020) and critical-thinking skills (Wang et al., 2017). It can also be added that block-based coding environments provide students with developing their creative thinking skills (Broll et al., 2018; Burker, 2015; Hainey et al., 2019; Iskrenovic-Momcilovic, 2019).

In the findings regarding interdisciplinary learning, benefits are acquired mostly in understanding mathematical concepts by integrating block-based coding tools (Akpinar & Aslan, 2015; Kalelioğlu, 2015; Ke, 2014; Ng & Cui, 2020; Sáez-López et al., 2019; Ortiz-Colon & Maroto Romo, 2016; Vidal-Silva et al., 2020).

Another finding during the research was about gender differences regarding cognitive abilities. According to the results, block-based coding environments positively affect female students' learning programming (Kalelioğlu, 2015; Papavlasopoulou et al., 2020; Puttick & Tucker-Raymond, 2018).

In the results regarding the affective domain, block-based coding had a positive effect on students' feelings (Armoni et.al, 2015; Basogain et al., 2018; Broll et al., 2018; Chou, 2019; Deng et al., 2019; Feijoo Garcia & De la Rosa, 2016;

Iskrenovic-Momcilovic, 2019; Mladenović et al., 2017; Papavlasopoulou et al., 2019; Papavlasopoulou et al., 2020; Papadakis & Orfanakis, 2018; Sáez-López et al., 2016, 2019; Sáez-López & Sevillano-García, 2017; Theodoropoulos et al., 2017; Weintrop & Wilensky, 2017). Notwithstanding the improvements, some issues were found regarding the females' interest and motivation towards programming with block-based coding tools (del Olmo-Muñoz et al., 2020; Gruenbaum, 2014).

Based on the findings, different factors were found that affect students' self-efficacy in block-based coding, both positively in terms of receiving coding education (Kim & Lee, 2017; Soykan & Kanbul, 2018) and negatively in terms of gender differences (Papavlasopoulou et al., 2020). Furthermore, there are improvements in positive attitudes towards learning to code (Deng et al., 2019; Erümit, 2020; Kalelioğlu, 2015; Maloney et al., 2010; Mladenović et al., 2017; Papavlasopoulou et al., 2019; Ortiz-Colon & Maroto Romo, 2016; Tangney et al., 2010; Wang et al., 2017). On the contrary, there are also negative changes found in students' attitudes toward computer science because of facing challenges on the tasks and lacking some cognitive skills like algorithmic thinking and problem-solving (Lambić et al., 2020; Zhao & Shute, 2019).

Implications for Practice

Based on the research articles examined, the following actions are suggested for implications for practice. In accordance with the study's findings, it was observed that the affective domain is rare when exploring the learning domain in research articles. Hence, future research can concentrate on the affective domain more. Additionally, there may be more research concentrating on critical-thinking skills regarding the results concerning the cognitive domain.

Besides, research articles that investigate the psychomotor domain can be done in further research, specifically for kindergarten. Since the number of studies with kindergarten students is very low, more research can be done with this grade level. This implication may help to find psychomotor domain in the research articles because younger students struggle with using computer items like keyboard and mouse.

In addition, more studies can be conducted for boosting female self-efficacy towards coding. Furthermore, the collaborative-learning approach might also be increased on further research besides the problem-based learning approach.

On another note, it was discovered that mainly the Scratch block-based coding tool could be promoted more in K-12 education because it is emphasized often in the research articles. Besides, the usage of underutilized block-based coding tools can be increased.

There may be more research in interdisciplinary work with different subject areas. It is seen that the research articles fundamentally concentrated in the field of Computer Science. Therefore, researching in other fields such as applied courses (Social Science including History and Geography, Art, Music, and Sport) can significantly contribute to the different educational areas.

Implications for Further Research

It is considered that the findings and results obtained will constitute a strong reference base for researchers and educators, and contribute to the field. Based on the research articles examined, implications for further research are given below.

Future research findings can be improved by making more comprehensive searches. It means similar researches can be repeated at specific periods, and the development in the field can be followed. Similar thematic research studies on coding can reveal the existing and new popular tools, and possible research topics in the study fields. Besides, this study dealt with students in K-12 education. Thus, other researchers can conduct studies considering higher education students and teachers. In addition, researchers can investigate the research articles that consider gaming in second life and physical devices supporting the learning of programming, such as tangible construction kits and educational robotics and applications for further research. In conclusion, it may be beneficial to include research articles and a diversity of various document types for review in future studies.

REFERENCES

Akkoyunlu, B. (2002). Educational technology in Turkey: Past, present, and future. *Educational Media International*, *39*(2), 165–174. https://doi.org/10.1080/09523980210155352

Alexiou-Ray, J., Raulston, C., Fenton, D., & Johnston, S. (2020). Coding: Coding in the k-12 classroom. In A. Ottenbreit-Leftwich & R. Kimmons (Eds.), *The K-12 educational technology handbook* (pp. 1-17). EdTech Books.

Anderson, L. W., & Krathwohl, D. R. (2001). *A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives*. Longman.

Armoni, M. (2016). Computing in schools computer science, computational thinking, programming, coding: The anomalies of transitivity in K-12 Computer Science Education. *ACM Inroads*, *7*(4), 24-27. https://dl.acm.org/doi/10.1145/3011071.

Barone, R. (2020). *What is block coding for kids?: All about drag-and-drop programming*. Id Tech. https://www.idtech.com/blog/what-is-block-based-coding

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, *2*(1), 48–54. https://doi.org/10.1145/1929887.1929905

Battelle for Kids. (2019). *Framework for 21st century learning.* https://www.battelleforkids.org/networks/p21/frameworks-resources

Ben-Ari, M. (1998). Constructivism in computer science education. *Proceedings of the Twenty-ninth SIGCSE Technical Symposium on Computer Science*

Education - SIGCSE '98, *30*(1), 257-261. https://doi.org/10.1145/273133.274308

Bers, M. U., González-González, C., & Armas–Torres, M. B. (2019). Coding as a playground: Promoting positive learning experiences in childhood classrooms. *Computers & Education*, *138*, 130–145. https://doi.org/10.1016/j.compedu.2019.04.013

Beyer, S. (2014). Why are women underrepresented in computer science? Gender differences in stereotypes, self-efficacy, values, and interests and predictors of future CS Course-taking and grades. *Computer Science Education, 24*(2-3), 153–192. https://doi.org/10.1080/08993408.2014.963363

Block-Based Coding (2020). In *Scratch Wiki*. https://en.scratch-wiki.info/wiki/Block-Based_Coding

Bloom, B. S., Krathwohl, D. R., & Masia, B. B. (1984). *Taxonomy of educational objectives: The classification of educational goals*. Longman.

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology, 3*(2), 77–101. https://doi.org/10.1191/1478088706qp063oa

Brookshear, J. G. (2006). *Computer science an overview*. Pearson Addison Wesley.

Çalık, M., & Sözbilir, M. (2014). Parameters of content analysis. *TED Eğitim ve Bilim, 39*(174). https://doi.org/10.15390/eb.2014.3412

Caulfield, J. (2021). *How to do thematic analysis*. Scribbr. https://www.scribbr.com/methodology/thematic-analysis/.

Code Week. (n.d.). *About.* https://codeweek.eu/about

Code.org. (n.d.). *Hour of code activities.* https://code.org/learn?grade=9%2B&subject=science

Codejig. (n.d.). *Block coding.* https://www.codejig.com/en/block-based-coding/

Common Sense Education. (2020a). *Best coding tools for elementary.* https://www.commonsense.org/education/top-picks/best-coding-tools-for-elementary

Common Sense Education. (2020b). *Best coding tools for high school students.* https://www.commonsense.org/education/top-picks/best-coding-tools-for-high-school-students

Common Sense Education. (2020c). *Best coding tools for middle school.* https://www.commonsense.org/education/top-picks/best-coding-tools-for-middle-school

Common Sense. (n.d.). *Common Sense is the leading nonprofit organization dedicated to helping all kids thrive in a world of media and technology.* https://www.commonsense.org/

Creswell, J. W. (2009). *Research design: Qualitative, quantitative, and mixed methods approaches.* SAGE Publications, Inc.

Creswell, J. W. (2014). *Educational research: Planning, conducting, and evaluating quantitative and qualitative research, Fourth Edition*. Pearson Education.

CS for ALL. (n.d.). *About CS for all.* https://www.csforall.org/about/

Demirbaş, A. (2019). *Kodlama eğitiminin değerlendirilmesi araştırması.* Yegitek. https://yegitek.meb.gov.tr/meb_iys_dosyalar/2020_03/26131559_kodlamaegitimindeYerlendirilmesiarastirmasi.pdf

Deng, W., Pi, Z., Lei, W., Zhou, Q., & Zhang, W. (2019). Pencil code improves learners' computational thinking and computer learning attitude. *Computer Applications in Engineering Education, 28*(1), 90-104. https://doi.org/10.1002/cae.22177C

European Schoolnet. (2015). *Computing our future: Computer programming and coding*. http://www.eun.org/documents/411753/817341/Computing+our+future_final_2015.pdf/d3780a64-1081-4488-8549-6033200e3c03

European Schoolnet. (2017). *Turkey country report on ICT in education.* http://www.eun.org/documents/411753/839549/Country+Report+Turkey+2017.pdf/054bdb93-3978-42bf-8040-ee590f9efe13

Ganimian, A. J., Vegas, E., & Hess, F. M. (2020). *Realizing the promise: How can education technology improve learning for all*. Brookings Institution.

Garneli, V., Giannakos, M. N., & Chorianopoulos, K. (2015). Computing education in K-12 schools: A review of the literature. *IEEE Global Engineering Education Conference (EDUCON).* https://ieeexplore.ieee.org/abstract/document/7096023/

Google. (2014). *Women who choose computer science–what really matters: The critical role of encouragement and exposure.* https://static.googleusercontent.com/media/edu.google.com/tr//pdfs/women-who-choose-what-really.pdf

Gretter, S., Yadav, A., Sands, P., & Hambrusch, S. (2019). Equitable learning environments in k-12 computing. *ACM Transactions on Computing Education*, *19*(3), 1-16. https://doi.org/10.1145/3282939

Hewitt, W. D., Leise, C., & Hall, A. (n.d.). *Affective domain*. PCrest. https://pcrest.com/research/fgb/2_3_6.pdf

Hour of Code. (2021). *FAQs.* https://hourofcode.com/tr/en

ISTE. (2016). *Standards for students.* https://www.iste.org/standards/for-students

Jain, S. (2021, April 9). *Why computer programming or coding is so important for students in today's era*. India Today. https://www.indiatoday.in/education-today/featurephilia/story/why-computer-programming-or-coding-is-so-important-for-students-in-today-s-era-1789171-2021-04-09

Januszewski, A., & Molenda, M. (2008). *Educational technology: A definition with commentary.* Routledge.

Johnson, L., Adams Becker, S., Estrada, V., & Freeman, A. (2014). *NMC Horizon Report: 2014 K-12 edition*. New Media Consortium (NMC). http://www.nmc.org/pdf/2014-nmc-horizon-report-he-EN.pdf

K–12 CS. (2016). *Computer science framework.* http://www.k12cs.org

Kalelioglu, F., Gulbahar, Y., Akcay, S., & Dogan, D. (2014). Curriculum integration ideas for improving the computational thinking skills of learners through programming via scratch. In *Local Proceedings of the 7th International Conference on Informatics in Schools: Situation, Evolution and Perspectives,* (pp.101-112).

Krathwohl, D. R., Bloom, B. S., & Masia, B. B. (1973). *Taxonomy of educational objectives: The classification of educational goals, handbook ii affective domain*. Longman.

Krippendorf, K. (2004). *Content analysis: An introduction to its methodology*. SAGE.

Lau, D. C.-M. (2001). Analysing the curriculum development process: Three models. *Pedagogy, Culture & Society*, *9*(1), 29–44. https://doi.org/10.1080/14681360100200107

Lye, S. Y., & Koh, J. H. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behaviour*, *41*, 51-61. https://doi.org/10.1016/j.chb.2014.09.012

Machanick, P. (2007). A social construction approach to computer science education. *Computer Science Education*, *17*(1), 1-20. https://doi.org/10.1080/08993400600971067

Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The scratch programming language and environment. *ACM Transactions on Computing Education*, *10*(4), 1–15. https://doi.org/10.1145/1868358.1868363

Master, A., Cheryan, S., & Meltzoff, A. N. (2016). Computing whether she belongs: Stereotypes undermine girls' interest and sense of belonging in computer science. *Journal of Educational Psychology*, *108*(3), 424-437. https://doi.org.10.1037/edu0000061

Miles, M. B., & Huberman, A. M. (1994). *Qualitative data analysis: An expanded sourcebook* (2nd ed.). Sage Publications.

Miliszewska, I., Barker, G., Henderson, F., & Sztendur, E. (2006). The issue of gender equity in computer science: What students say. *Journal of Information Technology Education: Research*, *5,* 107-120. https://doi.org.10.28945/237

MoNE- Ministry of National Education. (2018a). *Bilişim teknolojileri ve yazılım dersi öğretim programı (İlkokul 1, 2, 3 ve 4. Sınıflar).* http://mufredat.meb.gov.tr/Dosyalar/2018813171732131-4-201891%20Bili-%C5%9Fim%20Teknolojileri%20ve%20Yaz%C4%B1l%C4%B1m%20(1-4.%20S%C4%B1n%C4%B1flar).pdf

MoNE- Ministry of National Education. (2018b). *Bilişim teknolojileri ve yazılım dersi öğretim programı (Ortaokul 5 ve 6. Sınıflar).*

http://mufredat.meb.gov.tr/Dosyalar/2018124103559587Bili%C5%9Fim%20Teknolojileri%20ve%20Yaz%C4%B1l%C4%B1m%205-6.%20S%C4%%-B1n%C4%B1flar.pdf

MoNE- Ministry of National Education. (2018c). *Mesleki ve teknik anadolu lisesi. Anadolu meslek ve anadolu teknik programı. Bilişim teknolojileri alanı. Çerçeve öğretim programı*. http://mufredat.meb.gov.tr/Dosyalar/201918154454403-EK-1_B%C4%B0L%C4%B0%C5%9E%C4%B0M%20TEKNOLOJ%-C4%B0LER%C4%B0_%C3%87%C3%96P.pdf

MoNE- Ministry of National Education. (2019). *Öğrenciler kodlamayı "MASKOD" ile öğrenecek.* https://www.meb.gov.tr/ogrenciler-kodlamayi-maskod-ile-ogrenecek/haber/18710/tr

Moors, L., Luxton-Reilly, A., & Denny, P. (2018). Transitioning from block-based to text-based programming languages. *2018 International Conference on Learning and Teaching in Computing and Engineering (LaTICE).* https://doi.org/10.1109/LaTICE.2018.000-5

N.P.D. (2019, October 8). *According to the NPD group, 73 percent of U.S. consumers play video games.* https://www.npd.com/news/press-releases/2019/according-to-the-npd-group-73-percent-of-u-s-consumers-play-video-games/

Neuendorf, K. A. (2017). *The content analysis guidebook*. SAGE.

Nickerson, R. S. (1983). Computer programming as a vehicle for teaching thinking skills. *The Journal of Philosophy for Children, 4*(3), 42-48. https://doi.org/10.5840/thinking19834310

Noone, M., & Mooney, A. (2018). Visual and textual programming languages: A systematic review of the literature. *Journal of Computers in Education, 5*(2), 149–174. https://doi.org/10.1007/s40692-018-0101-5

OECD. (2020, June). *Education policy outlook Turkey.* https://www.oecd.org/education/policy-outlook/country-profile-Turkey-2020.pdf

Patson, N. D. (2020, May 29). *Getting students to discuss by channelling the affective domain*. Faculty Focus. *Higher Ed Teaching Strategies from Magna Publications.* https://www.facultyfocus.com/articles/teaching-and-learning/getting-students-to-discuss-by-channeling-the-affective-domain/

Polyviou, N. (2021). *Beginning computer programming for kids*. Primo Toys. https://www.primotoys.com/beginningcomputerprogramming/

Price, T. W., & Barnes, T. (2015). *Comparing textual and block interfaces in a novice programming environment* [Conference presentation abstract]. Proceedings of the Eleventh Annual International Conference on International Computing Education Research - ICER '15. Association for Computing Machinery, New York. https://doi.org/10.1145/2787622.2787712

Reddy, P. (2021). *Block-based and text-based coding for kids*. Codingal. https://www.codingal.com/blog/coding-for-kids/block-based-and-text-based-coding/

Resnick, M. (2012). Reviving Papert's dream. *Educational Technology*, *52*(4), 42-46.

Resnick, M., & Rusk, N. (1996). The computer clubhouse: Preparing for life in a digital world. *IBM Systems Journal*, *35*(3-4), 431-439. https://doi.org/10.1147/sj.353.0431

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

Rich, P. J., & Hodges, C. B. (2017). *Emerging research, practice, and policy on computational thinking*. Springer.

Rose, S. P., Habgood, M. P., & Jay, T. (2020). Designing a programming game to improve children's procedural abstraction skills in scratch. *Journal of Educational Computing Research*, *58*(7), 1372-1411. https://doi.org//10.1177/0735633120932871

Sáez-López, J., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two-year case study using "Scratch" in five schools. *Computers & Education*, *97*, 129-141. https://doi.org/10.1016/j.compedu.2016.03.003

Schiro, M. S. (2013). *Introduction to the curriculum ideologies. In curriculum theory: Conflicting visions and enduring concerns*. SAGE.

Sentance, S., Waite, J., & Kallia, M. (2019). Teaching computer programming with PRIMM: A sociocultural perspective. *Computer Science Education*, *29*(2-3), 136-176. https://doi.org/10.1080/08993408.2019.1608781

Simpson E.J. (1972). *The classification of educational objectives in the psychomotor domain*. Gryphon House.

Smith, M. K. (2000). *What is curriculum? Exploring theory and practice*. Infed. www.infed.org/biblio/b-curric.htm

Sobral, S. R. (2021). Bloom's taxonomy to improve teaching-learning in introduction to programming. *International Journal of Information and Education Technology*, *11*(3), 148–153. https://doi.org/10.18178/ijiet.2021.11.3.1504

Strawhacker, A., & Bers, M. U. (2018). What they learn when they learn coding: Investigating cognitive domains and computer programming knowledge in young children. *Educational Technology Research and Development*, *67*(3), 541–575. https://doi.org/10.1007/s11423-018-9622-x

Talan, T. (2020). Investigation of the studies on the use of scratch software in education. *Journal of Education and Future*. https://doi.org/10.30786/jef.556701

Tatar, C. & Eseryel, D. (2019). A literature review: Fostering computational thinking through game-based learning in K-12. *Association for Educational Communications and Technology*, 288-297.

Taylor, V. (2016). *Diversity gaps in computer science: Exploring the underrepresentation of girls, blacks and Hispanics*. Google. http://goo.gl/PG34aH

Temurnikar, W. (2020). *Importance of coding in 21st century education*. EdTechReview. https://edtechreview.in/trends-insights/insights/4264-importance-of-coding-in-21st-century-education.

Tissenbaum, M., & Ottenbreit-Leftwich, A. (2020). *A vision of K-12 computer science education for 2030*. Cacm. https://cacm.acm.org/magazines/2020/5/244329-a-vision-of-k-12-computer-science-education-for-2030/fulltext

Tucker, A., McCowan, D., Deek, F., Stephenson, C., Jones, J., & Verno, A. (2003). *A model curriculum for K–12 computer science: Report of the ACM K–12* *task force curriculum committee* (2nd ed.). Association for Computing Machinery.

Tynker. (2021). *Coding for kids: The why & how in 2021*. https://www.tynker.com/coding-for-kids/what-is-coding-for-kids

Vegas, E., & Fowler, B. (2020). *What do we know about the expansion of K-12 computer science education?* Brookings. https://www.brookings.edu/research/what-do-we-know-about-the-expansion-of-k-12-computer-science-education/

Vegas, E., Hansen, M., & Fowler, B. (2021). *Building skills for life: How to expand and improve computer science education around the world.* Brookings. https://www.brookings.edu/essay/building-skills-for-life-how-to-expand-and-improve-computer-science-education-around-the-world/

Webb, M., Davis, N., Bell, T., Katz, Y. J., Reynolds, N., Chambers, D. P., & Sysło, M. M. (2016). Computer science in K-12 school curricula of the 2LST century: Why, what and when? *Education and Information Technologies, 22*(2), 445–468. https://doi.org/10.1007/s10639-016-9493-x

Weinstein, J. (2021). *What is coding? Coding definition and uses*. Career Karma. https://careerkarma.com/blog/what-is-coding-used-for/

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

Westwood, P. S. (2008). *What teachers need to know about teaching methods*. ACER Press.

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

Wulf, T. (2005). *Constructivist approaches for teaching computer programming* [Conference presentation abstract]. Proceedings of the 6th Conference on Information Technology Education - SIGITE '05, Association for Computing Machinery, New York. https://doi.org/10.1145/1095714.1095771

Yildiz, S. N., Ates Cobanoglu, A., & Kisla, T. (2020). Development of the ICT teachers' perceptions of the contribution of scratch program to programming instruction scale. *International Journal of Computer Science Education in Schools*, *4*(1), 53–71. https://doi.org/10.21585/ijcses.v4i1.59

Appendix A

List of Articles

Akpinar, Y., & Aslan, Ü. (2015). Supporting children's learning of probability through video game programming. *Journal of Educational Computing Research*, *53*(2), 228–259. https://doi.org/10.1177/0735633115598492

Aksit, O., & Wiebe, E. N. (2019). Exploring force and motion concepts in middle grades using computational modeling: A classroom intervention study. *Journal of Science Education and Technology*, *29*(1), 65–82. https://doi.org/10.1007/s10956-019-09800-z

Arfé, B., Vardanega, T., & Ronconi, L. (2020). The effects of coding on children's planning and inhibition skills. *Computers & Education*, *148*, 103807. https://doi.org/10.1016/j.compedu.2020.103807

Armoni M., Meerbaum-Salant, O., & Ben-Ari, M. (2015). From Scratch to "real" programming. *ACM Transactions on Computing Education*. *14*(4). http://dx.doi.org/10.1145/2677087

Basogain, X., Olabe, M. Á., Olabe, J. C., & Rico, M. J. (2018). Computational thinking in pre-university blended learning classrooms. *Computers in Human Behavior*, *80*, 412–419. https://doi.org/10.1016/j.chb.2017.04.058

Broll, B., Lédeczi, Á., Zare, H., Do, D. N., Sallai, J., Völgyesi, P., Maróti, M., Brown, L., & Vanags, C. (2018). A visual programming environment for introducing distributed computing to secondary education. *Journal of Parallel and Distributed Computing*, *118*, 189–200. https://doi.org/10.1016/j.jpdc.2018.02.021

Burker, J. (2015). Going from bits to atoms: Programming in turtle blocks js and personal fabrication in youth maker projects. *Revista De Educación a Distancia (RED)*, (46). https://doi.org/10.6018/red/45/7

Çakıroğlu, Ü., & Mumcu, S. (2020). Focus-fight-finalize (3f): Problem-solving steps extracted from behavioral patterns in block based programming. *Journal of Educational Computing Research*, *58*(7), 1279–1310. https://doi.org/10.1177/0735633120930673

Çakiroğlu, Ü., Suiçmez, S. S., Kurtoğlu, Y. B., Sari, A., Yildiz, S., & Öztürk, M. (2018). Exploring perceived cognitive load in learning programming via scratch. *Research in Learning Technology*, *26*. https://doi.org/10.25304/rlt.v26.1888

Chang, C.H. (2019). Does the learning of computational thinking concepts interact with the practice of digital curation in children? A preliminary case study. *Journal of Educational Media & Library Sciences*, *56*(1), 45-68.

Chou, P.-N. (2019). Using ScratchJr to foster young children's computational thinking competence: A case study in a third-grade computer class. *Journal of Educational Computing Research*, *58*(3), 570–595. https://doi.org/10.1177/0735633119872908

Çiftci, S., & Bildiren, A. (2019). The effect of coding courses on the cognitive abilities and problem-solving skills of preschool children. *Computer Science Education*, *30*(1), 3–21. https://doi.org/10.1080/08993408.2019.1696169

del Olmo-Muñoz, J., Cózar-Gutiérrez, R., & González-Calero, J. A. (2020). Computational thinking through unplugged activities in early years of primary education. *Computers & Education*, *150*, 103832. https://doi.org/10.1016/j.compedu.2020.103832

Deng, W., Pi, Z., Lei, W., Zhou, Q., & Zhang, W. (2019). Pencil Code improves learners' computational thinking and computer learning attitude. *Computer Applications in Engineering Education*, *28*(1), 90–104. https://doi.org/10.1002/cae.22177

Denner, J., Werner, L., Campe, S., & Ortiz, E. (2014). Using game mechanics to measure what students learn from programming games. *International Journal of Game-Based Learning*, *4*(3), 13–22. https://doi.org/10.4018/ijgbl.2014070102

Erümit, A. K. (2020). Effects of different teaching approaches on programming skills. *Education and Information Technologies*, *25*(2), 1013–1037. https://doi.org/10.1007/s10639-019-10010-8

Estevez, J., Garate, G., & Grana, M. (2019). Gentle introduction to artificial intelligence for high-school students using scratch. *IEEE Access*, 7, 179027–179036. https://doi.org/10.1109/access.2019.2956136

Feijoo Garcia, P. G., & De la Rosa, F. (2016). RoBlock – Web app for programming learning. *International Journal of Emerging Technologies in Learning (IJET)*, *11*(12), 45. https://doi.org/10.3991/ijet.v11i12.6004

Fields, D., Vasudevan, V., & Kafai, Y. B. (2015). The programmers' collective: Fostering participatory culture by making music videos in a high school Scratch coding workshop. *Interactive Learning Environments*, *23*(5), 613–633. https://doi.org/10.1080/10494820.2015.1065892

Gruenbaum, P. (2014). Undergraduates teach game programming using scratch. computer, *47*(2), 82–84. https://doi.org/10.1109/mc.2014.49

Hainey, T., Baxter, G., & Ford, A. (2019). An evaluation of the introduction of games-based construction learning in upper primary education using a developed game codification scheme for scratch. *Journal of Applied Research in Higher Education*, *12*(3), 377–402. https://doi.org/10.1108/jarhe-02-2018-0031

Iskrenovic-Momcilovic, O. (2019). Pair programming with scratch. *Education and Information Technologies*, *24*(5), 2943–2952. https://doi.org/10.1007/s10639-019-09905-3

Jordonova, D., Koldova, H., Petraskova V., & Rosa, P. (2019). Interdisciplinary relations supporting propaedeutics of mathematics in primary education. *International Journal for Technology in Mathematics Education*, *26*(3), 151-157.

Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code.org. *Computers in Human Behavior*, *52*, 200–210. https://doi.org/10.1016/j.chb.2015.05.047

Ke, F. (2014). An implementation of design-based learning through creating educational computer games: A case study on mathematics learning during design and computing. *Computers & Education*, *73*, 26–39. https://doi.org/10.1016/j.compedu.2013.12.010

Kim, S., Choi, H., & Paik, S.-H. (2019). Using a systems thinking approach and a scratch computer program to improve students' understanding of the brønsted–lowry acid–base model. *Journal of Chemical Education*, *96*(12), 2926–2936. https://doi.org/10.1021/acs.jchemed.9b00210

Kim, S.W., & Lee, Y. (2017). A study of educational method using app inventor for elementary computing education. *Journal of Theoretical and Applied Information Technology*, *95*(18).

Lambić, D., Đorić, B., & Ivakić, S. (2020). Investigating the effect of the use of code.org on younger elementary school students' attitudes towards programming. *Behaviour & Information Technology*, 1–12. https://doi.org/10.1080/0144929x.2020.1781931

Lawanto, K., Close, K., Ames, C., & Brasiel, S. (2017). Exploring strengths and weaknesses in middle school students' computational thinking in scratch. *Emerging Research, Practice, and Policy on Computational Thinking*, 307–326. https://doi.org/10.1007/978-3-319-52691-1_19

Loganathan, P., Alwi, A.M., Romainor, N., Talib, C.A, Hanri, C., Malik, A.M.A., & Siang, K.H. (2019). Students' chemistry learning process through visual programming language: A preliminary study. *International Journal of Recent Technology and Engineering (IJRTE)*, *8*(1C2).

Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (M. (2013). Learning computer science concepts with Scratch. *Computer Science Education*, *23*(3), 239–264. https://doi.org/10.1080/08993408.2013.832022

Mladenović, M., Boljat, I., & Žanko, Ž. (2017). Comparing loops misconceptions in block-based and text-based programming languages at the K-12 level. *Education and Information Technologies*, *23*(4), 1483–1500. https://doi.org/10.1007/s10639-017-9673-3

Mladenović, M., Krpan, D., & Mladenović, S. (2017). Learning programming from Scratch. *Turkish Online Journal of Educational Technology*.

Mladenović, M., Žanko, Ž., & Mladenović, S. (2020). Impact of used programming language for K-12 students' understanding of the loop concept. *International Journal of Technology Enhanced Learning*, *12*(1), 79. https://doi.org/10.1504/ijtel.2020.10024760

Moreno-León, J., Robles G., & Román-González, M. (2015). Dr. scratch: Automatic analysis of scratch projects to assess and foster computational thinking. *RED-Revista de Educación a Distancia*. http://www.um.es/ead/red/46

Ng, O.-L., & Cui, Z. (2020). Examining primary students' mathematical problem-solving in a programming context: Towards computationally enhanced mathematics education. *ZDM – Mathematics Education*, *53*, 847–860. https://doi.org/10.1007/s11858-020-01200-7

Ortiz-Colon, A. M., & Maroto Romo, J. L. (2016). Teaching with scratch in compulsory secondary education. *International Journal of Emerging Technologies in Learning (IJET)*, *11*(02), 67. https://doi.org/10.3991/ijet.v11i02.5094

Papadakis, S., & Orfanakis, V. (2018). Comparing novice programing environments for use in secondary education: App Inventor for Android vs. Alice. *International Journal of Technology Enhanced Learning*, *10*(1/2), 44. https://doi.org/10.1504/ijtel.2018.10008587

Papavlasopoulou, S., Giannakos, M. N., & Jaccheri, L. (2019). Exploring children's learning experience in constructionism-based coding activities through design-based research. *Computers in Human Behavior*, *99*, 415–427. https://doi.org/10.1016/j.chb.2019.01.008

Papavlasopoulou, S., Sharma, K., & Giannakos, M. N. (2020). Coding activities for children: Coupling eye-tracking with qualitative data to investigate gender differences. *Computers in Human Behavior*, *105*, 105939. https://doi.org/10.1016/j.chb.2019.03.003

Pérez-Marín, D., Hijón-Neira, R., Bacelo, A., & Pizarro, C. (2020). Can computational thinking be improved by using a methodology based on metaphors and scratch to teach computer programming to children? *Computers in Human Behavior*, *105*, 105849. https://doi.org/10.1016/j.chb.2018.12.027

Puttick, G., & Tucker-Raymond, E. (2018). Building systems from scratch: An exploratory study of students learning about climate change. *Journal of Science Education and Technology*, *27*(4), 306–321. https://doi.org/10.1007/s10956-017-9725-x

Román-González, M., Pérez-González, J.-C., Moreno-León, J., & Robles, G. (2018). Can computational talent be detected? Predictive validity of the computational thinking test. *International Journal of Child-Computer Interaction*, *18*, 47–58. https://doi.org/10.1016/j.ijcci.2018.06.004

Rose, S., Habgood, M. P., & Jay, T. (2020). Designing a programming game to improve children's procedural abstraction skills in scratch. *Journal of Educational Computing Research*, *58*(7), 1372–1411. https://doi.org/10.1177/0735633120932871

Rose, S.P., Habgood, J.M.P. & Jay, T. (2017). An exploration of the role of visual programming tools in the development of young children's computational thinking. *The Electronic Journal of e-Learning*, *15(*4), 297-309.

Sáez-López, J.-M., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools. *Computers & Education*, *97*, 129–141. https://doi.org/10.1016/j.compedu.2016.03.003

Sáez-López, J.-M., & Sevillano-García, M.-L. (2017). Sensors, programming and devices in art education sessions. One case in the context of primary education / Sensores, programación y dispositivos en sesiones de Educación Artística. Un caso en el contexto de Educación Primaria. Cultura y Educación, *29*(2), 350–384. https://doi.org/10.1080/11356405.2017.1305075

Sáez-López, J.-M., Sevillano-García, M.-L., & Vazquez-Cano, E. (2019). The effect of programming on primary school students' mathematical and scientific understanding: educational use of mBot. *Educational Technology Research and Development*, *67*(6), 1405–1425. https://doi.org/10.1007/s11423-019-09648-5

Seralidou, E., & Douligeris, C. (2020). Learning programming by creating games through the use of structured activities in secondary education in Greece.

Education and Information Technologies, *26*(1), 859–898. https://doi.org/10.1007/s10639-020-10255-8

Soykan, F., & Kanbul, S. (2018). Analysing k12 students' self-efficacy regarding coding education. *TEM Journal*, *7*(1), 182-187. https://dx.doi.org/10.18421/TEM71-22

Tangney, B., Oldham, E., Conneely, C., Barrett, S., & Lawlor, J. (2010). Pedagogy and processes for a computer programming outreach workshop—the bridge to college model. *IEEE Transactions on Education*, *53*(1), 53–60. https://doi.org/10.1109/te.2009.2023210

Theodoropoulos, A., Antoniou, A., & Lepouras, G. (2017). How do different cognitive styles affect learning programming? Insights from a game-based approach in greek schools. *ACM Transactions on Computing Education*, *17*(1), 1–25. https://doi.org/10.1145/2940330

Tran, Y. (2018). Computational thinking equity in elementary classrooms: What third-grade students know and can do. *Journal of Educational Computing Research*, *57*(1), 3–31. https://doi.org/10.1177/0735633117743918

Vidal-Silva, C., Jiménez, C., Yupanqui, M.T, Sepúlveda, S., & Madariaga E. (2020). Development of algorithmic and mathematical-logic competences of children in chile with scratch. *International Journal of Scientific & Technology Research*. *9*(4).

Wang, X.-M., Hwang, G.-J., Liang, Z.-Y., & Wang, H.-Y. (2017). Enhancing students' computer programming performances, critical thinking awareness and attitudes towards programming: An online peer-assessment attempt. *Educational Technology & Society*, *20*(4), 58–68.

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

Werner, L., Denner, J., & Campe, S. (2015). Children programming games. *ACM Transactions on Computing Education*, *14*(4), 1–22. https://doi.org/10.1145/2677091

Wilson, A., Hainey, T., & Connolly, T. M. (2013). Using scratch with primary school children. *International Journal of Game-Based Learning*, *3*(1), 93–109. https://doi.org/10.4018/ijgbl.2013010107

Zhao, W., & Shute, V. J. (2019). Can playing a video game foster computational thinking skills? *Computers & Education*, *141*, 103633. https://doi.org/10.1016/j.compedu.2019.103633

Appendix B

List of Journals

Name of the Journals	f	%
ACM Transactions on Computing Education	4	6,78
Association for Educational Communications and Technology	1	1,69
Behaviour & Information Technology	1	1,69
Computer	1	1,69
Computer Applications in Engineering Education	1	1,69
Computer in Human Behavior	5	8,47
Computer Science Education	2	3,39
Computers & Education	5	8,47
Culture and Education	1	1,69
Education and Information Technologies	4	6,78
Educational Communications and Technology	1	1,69

Name of the Journals	f	%
IEEE	1	1,69
IEEE Transactions on Education	1	1,69
Interactive Learning Environments	1	1,69
International Journal of Scientific & Technology Research	1	1,69
International Forum of Educational Technology & Society	1	1,69
International Journal for Technology in Mathematics Education	1	1,69
International Journal of Child-Computer Interaction	1	1,69
International Journal of Emerging Technologies in Learning	2	3,39
International Journal of Game-Based Learning	2	3,39
International Journal of Recent Technology and Engineering	1	1,69
International Journal of Technology Enhanced Learning	2	2,39
Journal of Applied Research in Higher Education	1	1,69
Journal of Chemical Education	1	1,69

Name of the Journals	f	%
Journal of Educational Computing Research	5	8,47
Journal of Educational Media & Library Sciences	1	1,69
Journal of Parallel and Distributed Computing	1	1,69
Journal of Science Education and Technology	2	2,39
Journal of Theoretical and Applied Information Technology	1	1,69
RED- Revista de Educación a Distancia	2	3,39
Research in Learning Technology	1	1,69
TEM Journal	1	1,69
The Electronic Journal of e-Learning	1	1,69
Turkish Online Journal of Educational Technology	1	1,69
ZDM Mathematics Education	1	1,69
Total	59	100

Appendix C

Sample Code Book

id of the article	Name of the Journal	Title of the article	Author(s)	Year of the publication	Research Methods	Data Collection Instruments	The Education Level	Size of the Sample	Country	Subject Area	Programming tools	Teaching/ Learning Approach	Purpose	Theme(s)	Domain(s)	Brief Result
1																
2																
3																
4																
5																
6																
7																
8																
9																
10																
...																
59																