IJCSES

Volume 4, No: 4
April 2021
ISSN 2513-8359

International Journal of Computer Science Education In Schools

Editors

Dr Filiz Kalelioglu

Dr Yasemin Allsop

www.ijcses.org

International Journal of Computer Science Education in Schools

April 2021, Vol 4, No 4

DOI: 10.21585/ijcses.v4i4

Table of Contents

Developing and Assessing Computational Thinking in Secondary Education using a TPACK Guided Scratch Visual Execution Environment

Raquel Hijón Neira[1]

Miguel García-Iruela[1]

Cornelia Connolly[2]

[1]Universidad Rey Juan Carlos, Madrid, Spain

[2]National University of Ireland Galway, Ireland

DOI: 10.21585/ijcses.v4i4.98

Abstract

Effective and reliable assessment approaches to computational thinking in secondary education are in demand. This paper uses a guided technological pedagogical content knowledge (TPACK) framework, incorporating a visual execution environment (VEE) and Scratch project for secondary school students as a method to teach and assess computational thinking. The objective is to investigate if computational thinking and programming concepts can be improved upon following this method, and if the K-12 children are able to improve their computational thinking skills. The research study was conducted over 2 years in a school setting using the guided VEE and project developed following the dimensions of Computational Thinking process. The project participants came from two cohorts, an after-school programming camp and an in-school environment. Data was collected over two academic years and a quasi-experimental procedure with pre- and post-test was followed. The results demonstrate knowledge gain on computational and programming concepts and encourages us to convey how students translate (as opposed to transfer) their computational thinking experiences into reality. The results indicate the students achieved significant improvement in their computational thinking development.

Keywords: TPACK; Computational Thinking; Assessment; VEE; Scratch.

1. Introduction

Computational Thinking will be a fundamental skill used by everyone in the world (Wing, 2011) and is regarded as the thought processes, involving the formulation of problems and their solutions, characterised as computational steps and algorithms (Aho, 2012). Much research demonstrates how to incorporate computational thinking into classrooms (NRC, 2010; Weintrop et al., 2016; Yadav, Gretter, Good, & McLean, 2017) and many national curriculums are introducing computational thinking through a Computer Science curriculum at upper secondary school (Baron, Drot-Delange, Grandbastien, & Tort, 2014; Bell, Andreae, & Lambert, 2010; Brown, Sentance, Crick, & Humphreys, 2014). Coding is a key way to enable computational thinking (Lye & Koh, 2014) and development of related curriculum key in the enabling CT in secondary education and assessment.

Without attention to assessment, CT enactment in curriculum will have very little success (Grover & Pea, 2013). Assessment is intended to establish whether, and to what extent the curriculum intention has been achieved (Malone, 2011) and efforts to integrate and develop relevant CT-based assessments though developing are lacking (García-Peñalvo & Mendes, 2018; Grover & Pea, 2013). Assessment, and particularly formative assessment (Walsh & Dolan, 2009) help us to identify and bridge the gap between intended and the received curriculum. The two processes are not independent, but rather assessment follows the curriculum and at times dictates (Hargreaves, Earl, & Ryan, 1996).

This paper evaluates a TPACK Guided Scratch Visual Executing Environment for secondary school students as a method to teach, develop and assess computational thinking. The study contributes to the body of knowledge concerning development and assessment of computational thinking of visual programming (Brennan & Resnick, 2012), having developed a TPACK Guided Scratch VEE and the CT Knowledge Gain Test based on the environment. This work conveys how students translate (as opposed to transfer) their computational thinking experiences into reality. To appreciate the positioning of assessment of computational thinking in secondary education it is valuable to introduce the context and changes in the current literature and we begin by describing the literature and policy pertaining to such developments.

2. Literature Review

In 2006 Jeanette Wing published her article "Computational Thinking" (Wing, 2006) which is understood as fundamentally an analytical skill used to coordinate and interpret knowledge or data in order to accomplish various practical goals or tasks (NRC, 2010). Computational thinking should teach students to apply common CT elements to solve problems and discover new questions to explore within and across all disciplines (Hemmendinger, 2010). It was understood that coding is a key way to enable computational thinking (Lye & Koh, 2014) but CT may be applicable to a variety of unplugged problems that do not directly involve coding tasks (Wing, 2008). In 2011 Wing revisited the topic and provided a new definition “Computational thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent” (Wing, 2011, p. 3). It is important to focus on the importance of learners developing as computational creators (Resnick & Robinson, 2017) and such computational fluency involves not only an understanding of computational concepts and problem-solving strategies, but also the ability to create and express with and through digital technologies.

Computational thinking student learning and assessment is an area of development and studies have been quite varied. Some evaluated students engineering and programming skills as they debugged prebuilt faulty e-textile projects and their deconstruction, reverse engineering, and debugging skills (Fields, Searle, Kafai, & Min, 2012). Other studies presented a more systematic assessment of CT based science learning, using CTSiMe a Computational Thinking based science learning environment (Basu, Kinnebrew, & Biswas, 2014); the identification of CT patterns which young students abstract and develop during the creation of video-games in a controlled environment (Koh, Basawapatna, Bennett, & Repenning, 2010); the development in the student use of CS literacy from engaging in computationally rich activities provides an additional instrument for measuring the growth of CT (Grover, 2011). Moreno-León et al (2015) developed a web application to analyse automatically Scratch projects and provide feedback to improve programming and computational skills. SRI Education published reports providing principled approaches to designing assessment tasks which can generate valid evidence of students' abilities to think computationally exploring CS (Bienkowski, Snow, Rutstein, & Grover, 2015; Snow, Tate, Rutstein, & Bienkowski, 2017) and studies examined students usage of CT concepts and their awareness (Bower et al., 2017).

More recently, Lui et al (2019) demonstrated that CT literacy serves as a formative assessment tool, providing students with feedback benefiting their learning. However, Fields, Lui and Kafai (2019) presented findings revealing that assessment failed to capture the process of CT learning when they were learning with electronic textiles. Nevertheless, many studies highlight the benefits of CT and developing a CT integrated curriculum (Rich et al, 2019, Sung, 2019), CT-inspired teaching and learning tools (Grover 2017) and a CT-embedded learning environment (Muniz-Repiso, Caballero-Gonzálex, 2019).

2.1 Computational Thinking in Secondary Education

Coding is a key way to enable computational thinking (Lye & Koh, 2014) and so developing computer science and programming curriculum is key in the enabling CT integration in secondary education. The introduction of computing in secondary schools has been widely researched (Deek & Kimmel, 1999; Yadav, Gretter, Hambrusch, & Sands, 2016). In many countries the focus of computer science education at post-primary level has shifted from computer and ICT applications towards a more rigorous academic discipline (Bell et al., 2010; Brown et al., 2013; Hubwieser, 2012). The pattern of interest – a basic computing in the 1970s and 1980s, followed by a shift to digital literacies in the 1980s and 1990s, with a resurgence of interest in Computer Science in the past decade seems to match what has happened in the UK for example (Brown et al., 2013; Brown et al., 2014). The English national curriculum was changed in 2014, replacing Information Communication Technology (ICT) to a new subject of Computing which has more emphasis on computer science and programming principles, facilitating computational thinking (Csizmadia et al., 2015). New Zealand, similarly, introduced Computer Science in high schools nationally in 2011 (Bell, Andreae, & Robins, 2012). The revised NZ Computer Science curriculum content focuses on programming and gives students the chance to explore a range of computer science topics beyond programming, including algorithms and complexity, human-computer interaction, encryption, artificial intelligence, formal languages, computer graphics (Bell, Andreae, & Robins, 2014). In Ireland Computer Science was introduced as part of the curriculum in 2017 (NCCA, 2017) and a major component of their upper secondary specification is computational thinking. In Spain there is a computing curriculum in secondary education (BOE, 2015) and the subject "Technology, Programming and Robotics" has been taught from the 2014/15 academic year (INTEF, 2019). In reviewing the situation in Spain regarding Computing Education in pre-university stages made by the Spanish Computing Scientific Society (SCIE), with the support of the Spanish Board of Deans of Computing Schools (CODDI), it was recommended to establish a subject titled "Informatics", which was implemented as a mandatory course offered in both primary and secondary education (Velázquez-Iturbide, 2018). In many countries the focus of visual programming is primarily at primary level (Bell, Duncan, & Atlas, 2016; Duncan, Bell, & Atlas, 2017; Sáez-López, Román-González, & Vázquez-Cano, 2016).

Understanding the impact of a block-based programming environment in high school classrooms has been researched (Weintrop & Wilensky, 2017) and the work by Armoni et al (2015) focused on the transition from learning CS in middle school with Scratch, to learning CS in secondary school using a "real" programming language and a professional software development environment. Results demonstrated evidence to justify learning CS in middle schools, although there were no significant differences in achievements compared to students who had not studied Scratch (Armoni, Meerbaum-Salant, & Ben-Ari, 2015). This is consistent with the results of Levy et al. (2003), who showed that the use of the Jeliot program animation system primarily benefited the students who are capable of learning but not outstanding.

2.2 Serious Games and MaKey MaKey as a Teaching Resources

Game is understood as a playful action without a concrete, free and voluntary purpose. The win-lose dynamic is intrinsic. In this study, we use educational games (which have a specific purpose) where losing is a new opportunity to learn. Through the game, skills are developed to study the environment or specific problems and be creative looking for solutions (Granic, Lobel, & Engels, 2014), in this case computational thinking. Structuralist theory considers that the game establishes the way of seeing the world and thinking of the child as CT will be a new concept for many to discover. The absence of this "learn to think" prevents further learning from having depth (they are not reflexive) and therefore do not activate the emotional part that enables long-term learning (Piaget & Inhelder, 1999). On the other-hand the Fogg model (2009) designed to change human behaviour, establishes that three elements are necessary to modify the behaviour – motivation, skills, and a trigger. An educational game facilitates dynamics in which these three components converge simultaneously, being an optimal method for teaching-learning dynamics of new concepts, in our case concepts related to programming and the development of computational thinking.

MaKey MaKey was developed at MIT and is a simple hardware platform for improvising tangible user interfaces (Collective & Shaw, 2012). The Guided Scratch Visual Executing Environment (VEE) in this study can be used with a MaKey-MaKey device. The use of Makey-Makey is closely related to the constructivist conception of education, since it corresponds to the user, the design, construction and, where appropriate, modification of the controls to be used. Interactive controls have proven to be a good tool for the promotion of class interactivity (Álvarez Martínez & Llosa Espuny, 2010). In addition, the inclusion of the design of tangible game controls which develop inventiveness in Makey-Makey broadens learning opportunities (Lee et al., 2014). Martinez and Stager (2013) detail some of the main ideas that underlie the didactic use of this device:

- Learn by doing: You learn more when learning is part of something that is interesting. You learn best when we use what we learn to do something we really want.
- Technology as a building material: If you can use technology to do things, you can do more interesting things. And you can learn much more by doing them.
- Fun is not easy: We learn and work better if we enjoy what we are doing. But enjoying is not synonymous with easy. The greatest enjoyment occurs when the challenge is difficult.
- Learn to learn: The belief that you can only learn when someone teaches you is widespread. You don't always have someone who can teach you what you want to learn.
- Take the time to do the job: It is important to learn how to manage time for yourself to achieve the desired objectives.
- You cannot do well if you have not made a mistake: Complex things do not work at first.
- The only way to get it right is to analyze the problems that produced the previous failures.

Another important aspect of the MaKey-MaKey device is that it allows for easy adaptation to any need, and this was particularly important for our study in interfacing with the VEE and assessing computational thinking.

3. Research Design

This paper evaluates a TPACK Guided Scratch Visual Executing Environment for secondary school students as a method to teach, develop and assess computational thinking. The two research questions are as follows: Can computational thinking and programming concepts be improved with a TPACK Visual Execution Environment and Scratch on K-12 students? And secondly, by using this TPACK Guided Scratch VEE and Scratch are students able to improve their computational thinking skills?

3.1 Theoretical Foundation

The design of this study drew on the TPACK framework (Koehler & Mishra, 2009) in the integration of the necessary knowledge and the development of a useful tool to transmit the concepts of computational thinking which Scratch supports (Brennan & Resnick, 2012). The TPACK model defines the area in which technology is consistently integrated in teaching and the transfer of knowledge to the student is enhanced. The intersection of three fields of knowledge is that of Content Knowledge (concepts of computational thinking); Pedagogical Knowledge (exhibition and serious games) and thirdly Technological Knowledge (programming with scratch and development of web pages.) At the intersections of the fields of knowledge, less significant areas of partial knowledge are generated since they lack one of the areas.

According to Grover and Pea (2013), there is a consensus on the elements that should be included in a computational thinking curriculum, such as abstractions and generalizations of patterns, including models and simulations. The work carried out by the creators of Scratch was considered in the Creative Computing document (Brennan, Balch, & Chung, 2014) whose objective is to explore computational thinking by the Scratch programming language. This project is based on Brennan & Resnick (2012), which classifies computational thinking into three dimensions: Concepts, Practices and Computational Perspectives depicted in Table 1.

Table 1. Dimensions of Computational Thinking (Brennan & Resnick, 2012)

Computational Concepts (7):

- Sequences
- Loops
- Parallelism
- Events
- Conditional
- Operators
- Data

Computational Practices (4):

- **Incremental and iterative development**
- Test and debug
- Reuse and mix
- Abstract and modularize

Computational perspectives (3):

- Express yourself
- Connect
- Question

The use of metaphors for educational purposes consists of transferring a known concept from one object to another to which it provides a new notion or intuition. In this study attempts have been made to make metaphors evident and, where possible, graphic representations have been used to facilitate assimilation. Below is a list of the main metaphors used (see Table 2). In the case of the "Operator" concept, it has not been considered necessary to evoke a metaphor, since the notion of mathematical operator is widely extended.

Table 2. Metaphors of Computational Concepts

Concept	**Metaphor**
Sequence	Cooking recipe
Variable	Container with label
Conditional	Detour on the road
Loop	How a clock works
Event	Traffic light operation
Synchronization	Set the same time on two watches
Computational thinking	NIM game

3.2 TPACK Scratch Visual Execution Environment

The TPACK Scratch Visual Execution Environment has pre-established programs, which include the theory and practice corresponding to each of the proposed lessons. This separation allows the teacher different sequencing from the one proposed, if necessary. Since some concepts are supported by prior learning it is necessary (e.g., to explain the operation of conditionals, it is necessary to previously understand logical operators). The order of topics proposed are presented in Table 3.

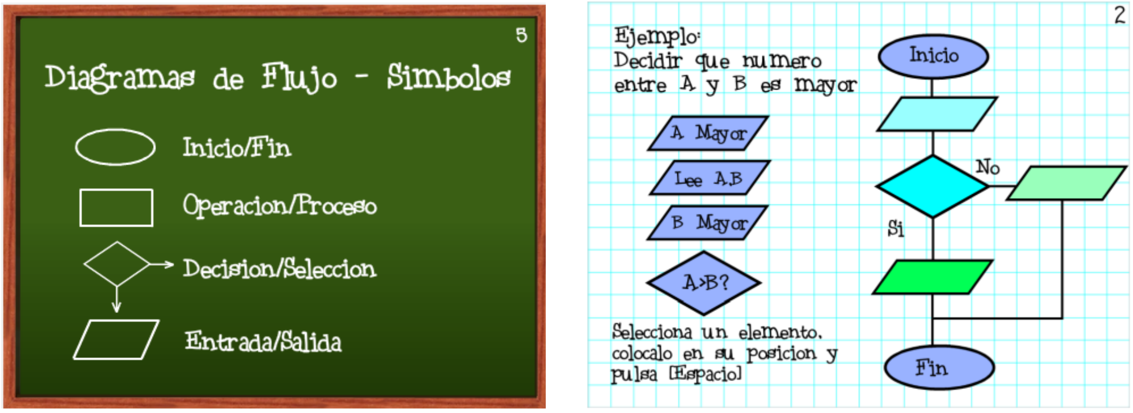
Table 3. Proposal for Sequencing Topics

Lesson 1. Sequences	Lesson 5. Loops
Lesson 2. Variables	Lesson 6. Events
Lesson 3. Operators	Lesson 7. Parallelism
Lesson 4. Conditionals	Lesson 8. Computational Thinking

The topics developed are closely related to the computational concepts implicit in Scratch, as a programming initiation language (Brennan & Resnick, 2012) and the first seven themes are valid tools for learning programming. The last concept includes the notion of computational thinking as an exercise of recapitulation and reinforcement of the previous points. Computational thinking is a complex competence that is related to the mental schemes of human beings, which allows to develop ideas and link abstraction (ideas-concepts) with pragmatism (action). It is not synonymous with programming, since it requires different degrees of abstraction and does not depend on computer equipment (unplugged). However, the use of computer equipment allows us to undertake tasks that without them would be unapproachable (Urbina Ramírez, 1999).

The TPACK Guided Scratch VEE has pre-established programs (Hernández Tijera & Perianes Rodriguez, 2018). It is a web application that allows accessing from any device apart from a PC (Smartphones, tablet, etc). It can be used with a MaKey-MaKey device or with the mouse and enables interaction at the students own pace. In order to access the Scratch applications embedded in the web pages, it is necessary to enable Flash Player and click on the green flag to start the different Scratch programs (http://scratch-tfm.000webhostapp.com/index.html).

In the development of each topic, the best way for the assimilation of the CT concept to be treated was considered and decided to first develop an exhibition and then carry out several practices. To highlight this separation, a visual key was used as a resource. The exhibition section has a classic slate background, and the practical part has a grid notebook background (see Figure 1). In the theoretical section, thanks to the interaction with the treated concept, students can interactively go at their own pace, allowing personalization to understand the concepts being exposed, developed around metaphors. In the practical section, thanks to the interaction with the treated concept, an instant feedback is showed, which allows both the assimilation and the accommodation of the new concepts (Figure 2).



5

Diagramas de Flujo - Simbolos

Inicio/Fin

Operacion/Proceso

Decision/Seleccion

Entrada/Salida

2

Ejemplo:
Decidir que numero
entre A y B es mayor

A Mayor

Lee A,B

B Mayor

A>B?

Selecciona un elemento,
colocalo en su posicion y
pulsa [Espacio]

Figure 1. Examples of exposure (left) and practice (right) in the TPACK Guided Scratch VEE

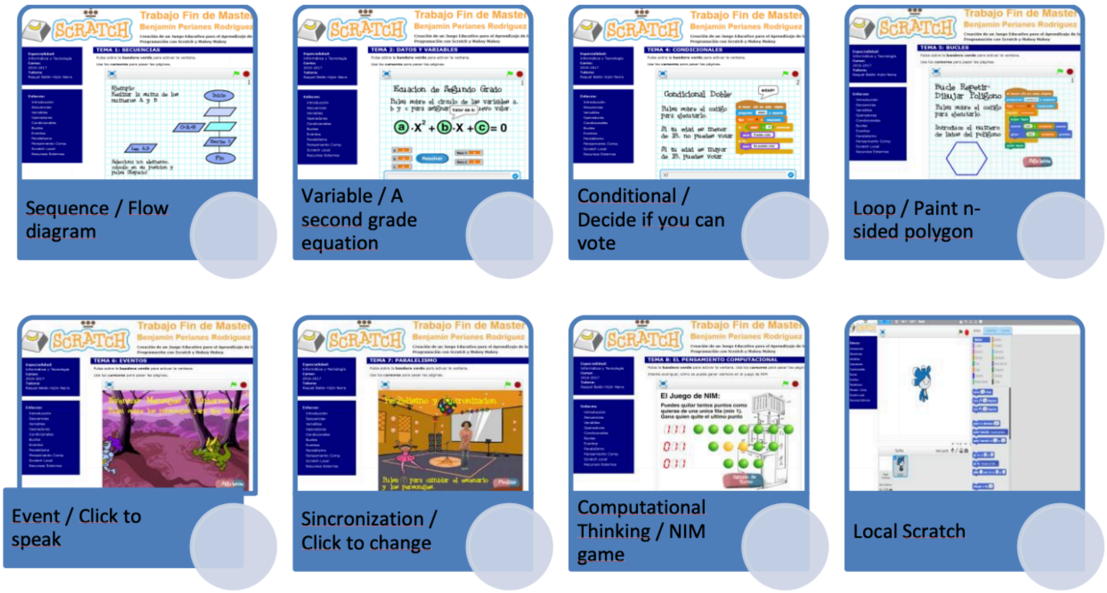

Sequence / Flow diagram
Variable / A second grade equation
Conditional / Decide if you can vote
Loop / Paint n-sided polygon
Event / Click to speak
Sincronization / Click to change
Computational Thinking / NIM game
Local Scratch

Figure 2. The TPACK Guided Scratch VEE with 7 computational Concepts and their serious games, last option is Local Scratch

3.3 Research Participants

The participants (N = 32) were K-12 students from two sites, some from an after-school programming camp (N = 6) and the others from a Madrid school (N = 26). The experience in both cases took place for 2 weeks and 6 hours per week, in total 12 hours. The distribution by gender is as follows: 54.8% girls and 45.2% boys. The experience took place in a computer room for 6 hours a week (2 hours a day) and no reward was given in grades or otherwise, the only reward was the students' own increasing motivation.

3.4 Data Collection

The experiment took place during the 2nd semester of two academic years, 2017-2018 and 2018-2019. A quasi-experimental procedure with pre- and post-test was followed. For the pre and post-tests, the same evaluation tests were used for concepts of computational thinking and programming on the one hand, and gains in computational thinking, on the other. Each test was completed individually by each student on their computer in class.

In each class, the students completed two pre-tests. The first was to verify what they knew about Brenan Resnick's 7 concepts of computational thinking (2012). The assessment consists of 12 free-text questions. These questions refer to the 7 computational Concepts of the first dimension of Computational Thinking (Sequence, variables, operators, conditionals, loops, events, parallelism, and the concept of computational thinking). The second test was to measure their computational thinking with a validated test (Román-González, Pérez-González, & Jiménez-Fernández, 2017). This test consisted of 28 multiple choice questions, with 4 possible options in each. Questions which cover the CT concepts of Basic directions & sequences, repeat times, repeat until, Simple conditional, complex conditional, while conditional and simple functions. The students then received 12 hours of class for 2 weeks, and they took the two post-tests to verify the improvements.

The tasks carried out by the students consisted of first an introduction to the theory and practice of each concept in the TPACK Guided Scratch VEE and then continuing working on it in Scratch. The tasks carried out were based on: algorithms, flow diagrams, operators, variables (Scratch's "ask" and "answer"), conditionals, loops, and then they worked with everything learned with a project we called "*Working Geometry with Scratch* ", students work on this project allowed them to incorporate the 2 dimensions of Brennan & Resnick (2012) left: Computational Practices (Incremental and iterative development, test and debug, reuse and mix, and abstract and modularize) and Computational perspectives (express yourself, connect and question). In this project the students had to make little programs that painted polygons, for example:

1. Draw an equilateral triangle of side 100
2. Draw a square of side 80
3. Draw a pentagon from side 50
4. Create a program that does the following:

- Set a suitable scenario and character (as if he were a school teacher)
- The character should say, “Hello, we are working with equilateral triangles. Equilateral triangles are those that have their three equal sides and their three angles measure 60º. I am going to draw the triangle that you want. How much do you want me to measure on your side?
- Then the program must draw a triangle aside the "answer" that the user enters on the keyboard.

5. We are going to do the same for a square.

- Modify the triangle program, it is very simple. Just change the number of laps and degrees.

6. The same for a pentagon
7. Same for n hexagon
8. Could you do it for a polygon with n sides?

4. Results and Discussion

4.1 Computational Thinking Concepts and Programming

A descriptive analysis of the results obtained is shown in Table 4, showing the minimum, maximum, mean and standard deviation of each test (pre and post). The results show that there are significant results in the results in the post-test. The minimum, maximum and mean values increase remarkably in the post-test results, although the dispersion increases minimally.

Table 4. Mean and Typical Deviation in The Test of Computational Thinking Concepts

	(n=32)			
	Min	**Max**	**Media**	**SD**
Pre	0,688	4,063	2,160	0,660
Post	6,125	9,750	8,867	0,931

The box-plots of the results in the evaluation of the Basic Computational Thinking Concepts, demonstrate the pre- and post-test, where each is delimited by the values Q1 (first quartile) and Q3 (third quartile). Each box groups 50% of the cases, highlighting the median. The lowest and highest value at the end of each diagram correspond to the values that are not less than Q1-1.5 · (Q3-Q1) and are not greater than Q3 + 1.5 · (Q3-Q1). The Analysis of variance of a factor (Anova) has been carried out to study the pre-test and the post-test and a value for F = 1105,31 and a p-value << 0,005 have been obtained, therefore it can be concluded that the data are significantly different. Comparing the pre-test with the post-test, after analysing the data, normality can be concluded for the study group (obtaining p> 0.05 significance using the Shapiro-Wilk test), allowing us to use the t-Student test for paired samples (p> 0.05 using bivariate correlation tests). In this test, it has been assumed that the null hypothesis can be established, since there are no differences between the means. Therefore, a p-value greater than 0.05 will reveal homogeneity in the samples. As a result, the difference between the pre-test and the post-test in the study of improvement in basic programming knowledge (t test analysis -6.707 and p-value 0.0001) and therefore, it is deduced that the students had a significant improvement in the test scores when following the course planning (p <0.0001). In order to collect additional information on the magnitude of the change produced in the students following the methodology explained in previous subsection, the size of the effect in the study group was calculated by means of the variation (Cohen, 1988), obtaining a g value = 8.4, corresponding to a very large effect (since it is> 0.5). According to these results, the students achieved a significant improvement in their global learning in the 7 concepts of Computational Thinking, the size of the effect is very large.

4.2 Computational Thinking Results for Each Concept

To verify which CT concepts, demonstrate the greater or less improvement achieved, we proceed to analyse which were more complicated or more affordable. The results show that there are significant results in the results in the post-test. The minimum, maximum and mean values increase remarkably in the post-test results, although the dispersion increases slightly in all concepts except for memory and sequence. Figure 3 shows the box-plots of the detailed results of each of the Computational Thinking Dimensions / concepts worked with Guided Scratch VEE in the pre- and post-test broken down by the 8 Dimensions studied (sequence, variables, operators, conditionals, loops, events, parallelism and the Computational Thinking Nim Game). Each box is delimited by the values Q1 (first quartile) and Q3 (third quartile). Each box groups 50% of the cases, highlighting the median, lowest and highest value at the end of each diagram corresponds to the values that are not less than Q1-1.5 · (Q3-Q1) and are not greater than Q3 + 1.5·(Q3-Q1). The Analysis of variance of a factor (Anova) has been carried out to study the pre-test and the post-test and a value for F = 1105,31 and a p-value of << 0,005 have been obtained, therefore it can be concluded that the data are significantly different. Comparing the pre-test with the post-test, after analysing the data, normality can be concluded for the study group (obtaining p> 0.05 significance using the Shapiro-Wilk test), allowing us to use the t-Student test for paired samples (p> 0.05 using bivariate correlation tests). In this test, it has been assumed that the null hypothesis can be established, since there are no differences between the means. Therefore, a p-value greater than 0.05 will reveal homogeneity in the samples.

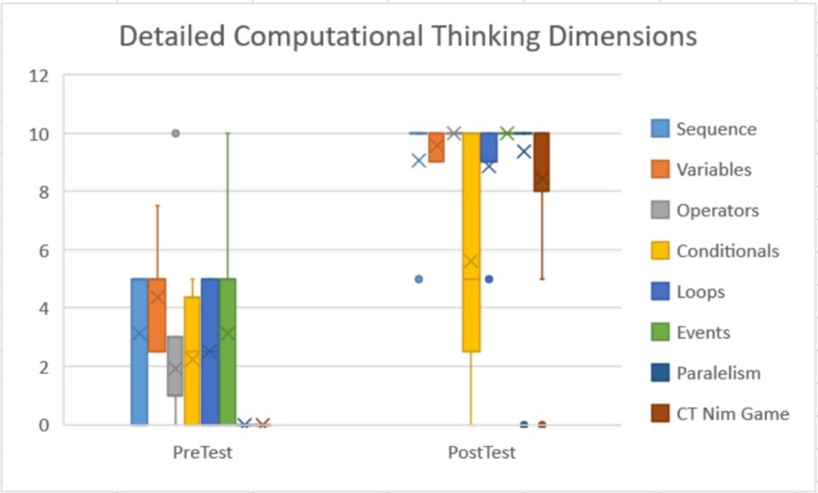


Figure 3. Box-plots for the group of students in pre- and post-tests in each of the CT dimensions worked with the TPACK Guided Scratch VEE

Table 5 shows the difference between the pre-test and the post-test in the study of improvement in each concept worked on following the procedure studied. Therefore, it is deduced that the students had a significant improvement in the knowledge of these dimensions of Computational Thinking at the end of the interaction (p <0.0001). In order to collect additional information on the magnitude of the change produced in the students following the TPACK Guided Scratch VEE methodology explained in previous subsection; the effect size in the study group was calculated by variation (Cohen, 1988), obtaining a value *variables* of g = 5,4 corresponding to a very large effect (since it is> 0.5), for *operators* of g = 6,9, corresponding to a very large effect (since it is> 0.5), for *conditionals* of g = 1,2, corresponding to a large effect (since it is> 0.5), for *loops* of g = 3,6, corresponding to a very large effect (since it is> 0.5), for *events* of g = 4 corresponding to a very large effect (since it is> 0.5), for *parallelism* of g = 7,6 corresponding to a very large effect (since it is> 0.5), for CT of g = 6,5 corresponding to a very large effect (since it is> 0.5).

Table 5. Study using t-Student and P-Value Analysis

	t test analysis	**p-value**
Sequence	-5,938	0,0001
Variables	-5,188	0,0001
Operators	-8,063	0,0001
Conditionals	-3,406	0,0001
Loops	-6,375	0,0001
Events	-6,875	0,0001
Paralelism	-9,375	0,0001
CT Nim Game	-8,438	0,0001

According to these results, learning is significant for all CT dimensions worked with the TPACK Guided Scratch VEE. At the beginning (pre-test) the newest or most unknown concepts for students are: *parallelism* and the concept of *computational thinking*. On the other hand, the most familiar concepts are that of *sequence, loops* and *events,* although they do not yet dominate. At the end of the intervention, all the dimensions have achieved a significant improvement, we can say that the concepts of *sequence, operators, events* and *parallelism* dominate; The rest of the concepts (*variables, loops, computational thinking*) are dominated by more than 80% of the group of participating students, and in *conditionals* it is where there is more dispersion, achieving more than 50% of the class to overcome it as well. On the other hand, the concepts with the greatest effect on learning are (in that order): *parallelism, operators, computational thinking, variables, loops, events, sequence,* and *conditionals*.

4.3 Assessment of Computational Thinking

With regards to assessment of computational thinking we firstly provide a descriptive analysis of the results obtained. Table 6 shows the minimum, maximum value, mean and standard deviation of each test (pre and post). The results show that there are significant results in the results in the post-test. The minimum, maximum and mean values increase remarkably in the post-test results, although the dispersion increases minimally.

Table 6. Mean and Typical deviation in the assessment of Computational Thinking

	(n=32)			
	Min	**Max**	**Media**	**SD**
Pre	3,929	7,500	4,576	0,987
Post	4,643	7,500	6,295	0,653

The box-plots of the results in the evaluation of the Computational Thinking Test in the pre- and post-test and the analysis of variance of a factor (Anova) has been carried out to study the preTest and the postTest and a value for F = 67,49 and a p-value < 0,005 have been obtained, therefore it can be concluded that the data are significantly different. Comparing the pre-test with the post-test, after analyzing the data, normality can be concluded for the study group (obtaining p> 0.05 significance using the Shapiro-Wilk test), allowing us to use the t-Student test for paired samples (p> 0.05 using bivariate correlation tests). In this test, it has been assumed that the null hypothesis can be established, since there are no differences between the means. Therefore, a p-value greater than 0.05 will reveal homogeneity in the samples. The t-test analysis is -1.719 and therefore the difference between the pre-test and the post-test in the study of improvement in basic programming knowledge. Therefore, it is deduced that the students had a significant improvement in the test scores when following the course planning (p <0.0001). In order to collect additional information on the magnitude of the change produced in the students following the methodology explained in previous subsection, the size of the effect in the study group was calculated by means of the variation in Cohen's D (Cohen, 1988), obtaining a g value = 2,1 corresponding to a very large effect (since it is> 0.5). According to these results, the students achieved a significant improvement in their Computational Thinking, the size of the effect is very large.

4.4 Assessment Computational Thinking Test by Concepts

In order to determine if by the intervention students improved differently and how on their Computational Thinking, we proceed to study the concepts assessed to determine which had the most improvement and which ones had the least. The results show that there are significant results in the results in the post-test. The minimum, maximum and mean values increase remarkably in the post-test results, although the dispersion increases slightly in all concepts except memory and sequence. Figure 4 shows the box-plots of the detailed results on the Computational Thinking Test exploited by concepts (*basic directions & sequences, repeat times, repeat until, simple conditional, complex conditional, while conditional, simple functions*). The Analysis of variance of a factor (Anova) has been carried out to study the preTest and the postTest and a value for F = 67,49 and a p-value of << 0,005 have been obtained, therefore it can be concluded that the data are significantly different.

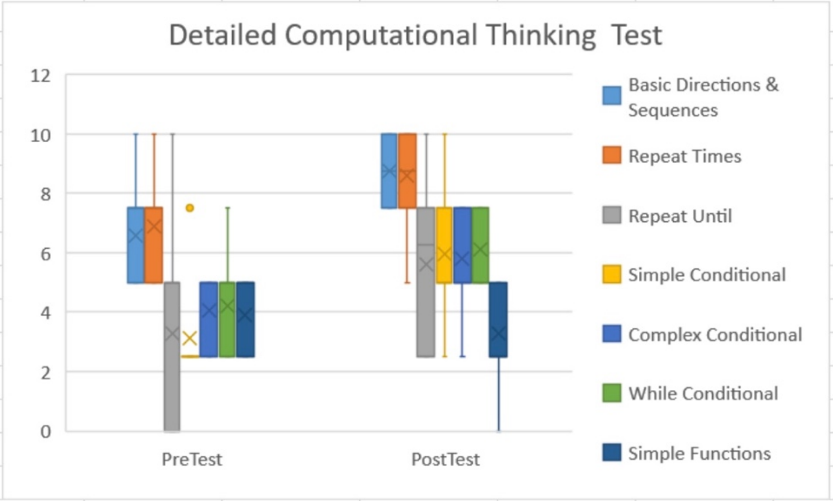


Figure 4. Box-plots for the group of students in pre- and post-tests in each CT concept assessed

Comparing the pre-test with the post-test, after analyzing the data, normality can be concluded for the study group (obtaining p> 0.05 significance using the Shapiro-Wilk test), allowing us to use the t-Student test for paired samples (p> 0.05 using bivariate correlation tests). In this test, it has been assumed that the null hypothesis can be established, since there are no differences between the means. Therefore, a p-value greater than 0.05 will reveal homogeneity in the samples. It is deduced that the students had a significant improvement on these concepts of Computational Thinking at the end of the interaction (p <0.0001) but for Simple Functions (p<0,018).

In order to collect additional information on the magnitude of the change produced in the students on each concept, the effect size in the study group was calculated by variation in Cohen's D (Cohen, 1988), obtaining a value for *basic directions & sequence* of g = 1,5, corresponding to a very large effect (since it is> 0.5), for *repeat times* of g = 1.06, corresponding to a very large effect (since it is> 0.5), for *repeat until* of g = 0.79, corresponding to a large effect (since it is> 0.5), for *simple conditional* of g = 1.54, corresponding to a very large effect (since it is> 0.5), for *complex conditional* of g = 1.16 corresponding to a very large effect (since it is> 0.5), for *while conditional* of g = 1.37 corresponding to a very large effect (since it is> 0.5).

5. Conclusion

There is worldwide interest on finding ways to improve students Computational Thinking skills and ways to assess it. This project uses a guided TPACK framework, incorporating a Scratch VEE for secondary school students as a method to teach and assess computational thinking. The objective was to investigate if computational thinking and programming concepts could be improved upon in applying this approach and if the K-12 children are able to improve their computational thinking skills.

The use of the serious games in the Guided Scratch VEE enables creativity in looking for solutions, similar to previous studies (Granic, Lobel, & Engels, 2014). The possibility of using a Makey-Makey can promote class interactivity (Álvarez Martínez & Llosa Espuny, 2010) and learning opportunities (Lee et al., 2014). Moreover, metaphors have been used to explain concepts (*Pérez-Marín, et al., 2020*; Pérez-Marín, Hijón-Neira, Martín-Lope, 2018), as well as abstractions and generalization of patterns (**Grover & Pea 2013)** presenting them as pre-established programs to guide the student in their learning. A balance of theory and practice, combined with the ordering of topics, which follows the sequence proposed in studies to teach CT (Brennan & Resnick, 2012). Furthermore, adopting user centred design (UX) the exhibition section and the practical section have different backgrounds, offering instant feedback. The experiment also took part over two academic years aligning to the validated test (Román-González, Pérez-González & Jiménez-Fernández, 2017). As mentioned previously, prior to the tasks being carried out by students, they first developed understanding of each concept in the TPACK Guided Scratch VEE and the continued their learning with Scratch, enabling them incorporate the 2 dimensions of computational practices and computational perspectives (Brennan & Resnick, 2012)

The first Research Question, RQ1, asked: Can Computational Thinking and programming concepts be improved with a Visual Execution Environment and Scratch on K-12 students? In concluding the study, it has been observed that a TPACK Guided Scratch VEE, and an iterative incremental project based on polygons, that the students created in Scratch at their own pace, incorporating computational practices (*incremental and iterative*, *test and debug*, *reuse, and mix*, and *abstract and modularize*) and computational perspectives (*express yourself, connect and question*). The students achieved significant improvement in their learning of the concepts of Computational Thinking. The students achieved a significant improvement in their global learning in the seven concepts of Computational Thinking, the size of the effect is very large. According to these results, learning is significant for all CT dimensions worked with the TPACK Guided Scratch VEE. At the beginning (pre-test) the newest or most unknown concepts for students are: *parallelism* and the concept of *computational thinking*. On the other hand, the most familiar concepts are that of *sequence, loops,* and *events*, although they do not yet dominate. At the end of the intervention, all the dimensions have achieved a significant improvement, we can say that the concepts of *sequence, operators, events,* and *parallelism* dominate; The rest of the concepts (*variables, loops, computational thinking*) are dominated by more than 80% of the group of participating students, and in *conditionals* it is where there is more dispersion, achieving more than 50% of the class to overcome it as well. On the other hand, the concepts with the greatest effect on learning are (in that order): *parallelism, operators, computational thinking, variables, loops, events, sequence,* and *conditionals*.

The second Research Question, RQ2 investigated if by using this TPACK Guided Scratch VEE are students able to improve their computational thinking skills? The students achieved a significant improvement in their Computational Thinking and the size of the effect is very large. According to these results, learning is significant for all CT concepts the test measured but for *simple functions*. At the beginning (pre-test) the newest or most unknown concepts for students are: *repeat until, simple* and *complex conditionals, while conditional* and *simple functions*. As in previous analysis, the most familiar concepts are that of *basic directions & sequence* in the first place, followed by *repeat times*, they are all about to pass. At the end of the intervention, all the concepts but *functions* have achieved a significant improvement, we can say that the concepts of *basic directions & sequence* and *repeat times* students got it excellent; the rest of the concepts (*repeat until, simple* and *complex conditional, while conditional* are also well understood. On the other hand, the concepts with the greatest effect on learning are (in that order): *simple conditions, basic directions, while conditional, complex conditionals, complex conditional, repeat times, repeat until* and *simple functions*.

The results obtained on K-12 students for a total of 12 hours of classes using the TPACK Guided Scratch VEE and developing guided projects with Scratch, the results of this experiment demonstrate that students gained a significant improvement in the test scores when following the course planning. According to these results, the students achieved a significant improvement in their global learning in the seven concepts of computational thinking, the size of the effect is very large. Therefore, the use of such VEE provides an aid to the teacher in introducing the CT concepts, and provides guidance specific to metaphors and serious games. It provides a better introduction to just starting with Scratch from Scratch, offering well established steps for explaining abstract concepts to young students.

When studying the results for each specific dimension, it is deduced that students had a significant improvement in the knowledge of all dimensions of computational thinking at the end of the interaction, and that is a good way for explaining such new concepts to students as parallelism or computational thinking. Since at the end of the intervention, all dimensions have achieved a significant improvement, we can say that using the Guided TPACK VEE and Scratch we can help teachers teach such complex and novel concepts such as: Sequence, Variables, operators, conditionals, loops, events, parallelism, and the CT concept itself.

It has been proven that there is a significant improvement in students Computational Thinking. The magnitude of change produced by students on each concept is a very large for all and ordered form more to less are: basic directions & sequence, repeat times, repeat until, simple conditional, complex conditional, while conditional. Therefore, by using the Guided TPACK VEE and Scratch teachers can teach CT concepts in the classroom more smoothly and it will reduce preparation time providing very significantly outcomes for their students.

While the results demonstrate a definite statement on improvement of CT skills when a TPACK Guided Scratch VEE was used, the work is specific to one geographic location and pedagogic approach adopted. The work, being a quasi-experimental research case study with one school and after school context, has the primary limitation of a narrow focus. While such an approach does not facilitate the development of generalisations, it can effectively point out possible results, which require further investigation and validation.

This paper presented a study carried out with K-12 secondary students. A procedure was followed where the computational concepts, practices, and perspectives, according to a TPACK Guided Scratch VEE were adopted. The class built a project on Scratch where the concepts only worked combining computational thinking perspectives and practices. To assess the computational thinking two tests were carried out to answer to our research questions and the results demonstrate knowledge gained on computational and programming concepts. The findings demonstrate that students managed to make complex programs combining what they learned in an incremental way, which provides insight to CS educators in pedagogical approaches.

References

Aho, A. V. (2012). Computation and computational thinking. *The Computer Journal, 55*(7), 832-835. https://doi.org/10.1093/comjnl/bxs074

Álvarez Martínez, C., & Llosa Espuny, J. (2010). *Formative evaluation with quick feedback using interactive controls.* Paper presented at the XVI Conference on University Teaching of Computing, University of Santiago de Compostela. Escola Técnica Superior d'Enxeñaría.

Armoni, M., Meerbaum-Salant, O., & Ben-Ari, M. (2015). From scratch to "real" programming. *ACM Transactions on Computing Education (TOCE), 14*(4), 1-15. https://doi.org/10.1145/2677087

Baron, G. L., Drot-Delange, B., Grandbastien, M., & Tort, F. (2014). Computer science education in French secondary schools: Historical and didactical perspectives. *ACM Transactions on Computing Education (TOCE), 14*(2), 1-27. https://doi.org/10.1145/2602486

Basu, S., Kinnebrew, J. S., & Biswas, G. (2014). Assessing student performance in a computational-thinking based science learning environment. In *International conference on intelligent tutoring systems* (pp. 476-481): Springer.

Bell, T., Andreae, P., & Lambert, L. (2010). *Computer science in New Zealand high schools.* Paper presented at the 12th Australasian Conference on Computing Education, Brisbane, Australia.

Bell, T., Andreae, P., & Robins, A. (2012). *Computer science in NZ high schools: the first year of the new standards.* . Paper presented at the 43rd ACM technical symposium on Computer Science Education.

Bell, T., Andreae, P., & Robins, A. (2014). A case study of the introduction of computer science in NZ schools. *ACM Transactions on Computing Education (TOCE), 14*(2), 1-31. https://doi.org/10.1145/2602485

Bell, T., Duncan, C., & Atlas, J. (2016). *Teacher feedback on delivering computational thinking in primary school.* Paper presented at the 11th Workshop in Primary and Secondary Computing Education. https://doi.org/10.1145/2978249.2978266

Bienkowski, M., Snow, E., Rutstein, D., & Grover, S. (2015). *Assessment design patterns for computational thinking practices in secondary computer science: A first look.* Retrieved from https://pact.sri.com/downloads/Assessment-Design-Patterns-for-Computational%20Thinking-Practices-Secondary-Computer-Science.pdf

BOE. (2015). *Orden ECD/65/2015 por la que se describen las relaciones entre las competencias, los contenidos y los criterios de evaluación de la Educación Primaria, la Educación Secundaria Obligatoria y el Bachillerat.*

Bower, M., Wood, L. N., Lai, J. W., Highfield, K., Veal, J., Howe, C., ... & Mason, R. (2017). Improving the computational thinking pedagogical capabilities of school teachers. *Australian Journal of Teacher Education*, *42*(3), 53-72.

Brennan, K., Balch, C., & Chung, M. (2014). *Creative computing: Scratch curriculum guide*. Retrieved from https://creativecomputing.gse.harvard.edu/guide/

Brennan, K., & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking.* Paper presented at the American educational research association, Vancouver, Canada.

Brown, N. C., Kölling, M., Crick, T., Peyton Jones, S., Humphreys, S., & Sentance, S. (2013). *Bringing computer science back into schools: Lessons from the UK.* Paper presented at the 44th ACM Technical Symposium on Computer Science Education (SIGCSE'13). New York. https://doi.org/10.1145/2445196.2445277

Brown, N. C., Sentance, S., Crick, T., & Humphreys, S. (2014). Restart: The resurgence of computer science in UK schools. *ACM Transactions on Computing Education (TOCE), 14*(2), 9. https://doi.org/10.1145/2602484

Cohen, J. (1988). *Statistical Power Analysis for the Behavioural Sciences*. Hillsdale, NJ, USA: Erlbaum.

Collective, B. M., & Shaw, D. (2012). *Makey Makey: improvising tangible and nature-based user interfaces.* Paper presented at the Sixth International Conference on Tangible, Embedded and Embodied Iinteraction.

Deek, F. P., & Kimmel, H. (1999). Status of computer science education in secondary schools: One state's perspective. *Computer Science Education, 9*(2), 89-113. https://doi.org/10.1076/csed.9.2.89.3808

Duncan, C., Bell, T., & Atlas, J. (2017). *What do the teachers think? Introducing computational thinking in the primary school curriculum.* Paper presented at the Nineteenth Australasian Computing Education Conference.

Fields, D. A., Searle, K. A., Kafai, Y. B., & Min, H. S. (2012). *Debuggems to assess student learning in e-textiles.* Paper presented at the 43rd ACM technical symposium on Computer Science Education. https://doi.org/10.1145/2157136.2157367

Fields, D. A., Lui, D., & Kafai, Y. B. (2019). Teaching computational thinking with electronic textiles: Modeling iterative practices and supporting personal projects in exploring computer science. In *Computational thinking education* (pp. 279-294). Springer, Singapore.

García-Peñalvo, F. J., & Mendes, A. J. (2018). Exploring the computational thinking effects in pre-university education. *Computers in Human Behavior, 80*, 407-411. https://doi.org/10.1016/j.chb.2017.12.005

García-Valcárcel-Muñoz-Repiso, A., & Caballero-González, Y. A. (2019). Robotics to develop computational thinking in early Childhood Education. *Comunicar. Media Education Research Journal*, *27*(1).

Granic, I., Lobel, A., & Engels, R. C. (2014). The benefits of playing video games. *American psychologist, 69*(1), 66. https://doi.org/10.1037/a0034857

Grover, S. (2011). *Robotics and engineering for middle and high school students to develop computational thinking.* Paper presented at the American Educational Research Association, New Orleans, LA.

Grover, S., & Pea, R. (2013). Computational Thinking in K–12: A Review of the State of the Field. *Educational Researcher, 42*(2), 38-43. https://doi.org/10.3102/0013189X12463051

Grover, S. (2017). Assessing algorithmic and computational thinking in K-12: Lessons from a middle school classroom. In *Emerging research, practice, and policy on computational thinking* (pp. 269-288). Springer, Cham.

Hargreaves, A., Earl, L. M., & Ryan, J. (1996). *Schooling for change: Reinventing education for early adolescents*: Routledge.

Hemmendinger, D. (2010). A Plea for Modesty. *ACM Inroads, 1*(2), 4-7. https://doi.org/10.1145/1805724.1805725

Hernández Tijera, I., & Perianes Rodriguez, B. (2018). *Camino hacia la excelencia. Premios Trabajo Fin de Máster en Formación del profesorado de Educación Secundaria.* Retrieved from Colegio Oficial de Docentes. Sial Pigmalión.

Hubwieser, P. (2012). Computer science education in secondary schools: The introduction of a new compulsory subject. *Transactions in Computing Education, 12*(4), 161-164. http://dx.doi.org/10.1145/2382564.2382568.

INTEF. (2019). *Programación, robótica y pensamiento computacional en el aula: Situación en España*.

Koehler, M. J., & Mishra, P. (2009). What is technological pedagogical content knowledge? *Contemporary Issues in Technology & Teacher Education, 9*(1), 60-70.

Koh, K. H., Basawapatna, A., Bennett, V., & Repenning, A. (2010). Towards the automatic recognition of computational thinking for adaptive visual language learning. *IEEE Symposium on Visual Languages and Human-Centric Computing*, 59-66. https://doi.org/10.1109/VLHCC.2010.17

Lee, E., Kafai, Y. B., Vasudevan, V., & Davis, R. L. (2014). Playing in the arcade: Designing tangible interfaces with MaKey MaKey for Scratch games. In *Playful user interfaces* (pp. 277-292). Singapore: Springer.

Levy, R. B. B., Ben-Ari, M., & Uronen, P. A. (2003). The Jeliot 2000 program animation system. *Computers & Education, 40*(1), 1-15. https://doi.org/10.1016/S0360-1315(02)00076-3

Lui, D., Walker, J. T., Hanna, S., Kafai, Y. B., Fields, D., & Jayathirtha, G. (2020). Communicating computational concepts and practices within high school students' portfolios of making electronic textiles. *Interactive Learning Environments*, *28*(3), 284-301.

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

Malone, R. (2011). Curriculum Studies. In B. Walsh (Ed.), *Education Studies in Ireland: the Key Disciplines*. Dublin: Gill & Macmillan Ltd.

Martinez, S. L., & Stager, G. (2013). *Invent to learn. Making, Tinkering, and Engineering in the Classroom.* Paper presented at the Construting Modern Knowledge., Torrance, Canada.

Moreno-León, J., & Robles, G. (2015). *Dr. Scratch: A web tool to automatically evaluate Scratch projects.* Paper presented at the Workshop in primary and secondary computing education.

Moreno-León, J., Robles, G., & Román-González, M. (2015). Dr. Scratch: Automatic analysis of scratch projects to assess and foster computational thinking. *Revista de Educación a Distancia, 46*, 1-23.

NCCA. (2017). *Leaving Certificate Computer Science Specification*. Retrieved from http://ncca.ie/en/Curriculum_and_Assessment/Post-Primary_Education/Senior_Cycle/Consultation/LC-Computer-Science.pdf

NRC. (2010). *Report of a Workshop on the Scope and Nature of Computational Thinking* National Research Council, Washington DC: National Academies Press.

Pérez-Marín, D., Hijón-Neira, R., & Martín-Lope, M. (2018). A methodology proposal based on metaphors to teach programming to children. *IEEE Revista Iberoamericana de tecnologias del aprendizaje*, *13*(1), 46-53.

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.

Piaget, J., & Inhelder, B. (1999). *The Child's Conception of Space*. UK: Routledge.

Resnick, M., & Robinson, K. (2017). *Lifelong kindergarten: Cultivating creativity through projects, passion, peers, and play*: MIT Press.

Rich, K. M., Spaepen, E., Strickland, C., & Moran, C. (2020). Synergies and differences in mathematical and computational thinking: Implications for integrated instruction. *Interactive Learning Environments*, *28*(3), 272-283.

Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior, 72*, 678-691. https://doi.org/10.1016/j.chb.2016.08.047

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*, 128-141. https://doi.org/10.1016/j.compedu.2016.03.003

Snow, E., Tate, C., Rutstein, D., & Bienkowski, M. (2017). *Assessment design patterns for computational thinking practices in exploring computer science*. Retrieved from https://pact.sri.com/downloads/AssessmentDesignPatternsforComputationalThinking%20PracticesinECS.pdf

Sung, E. (2019). Fostering computational thinking in technology and engineering education: an unplugged hands-on engineering design approach. *Technology and Engineering Teacher*, *78*(5), 8-13.

Urbina Ramírez, S. (1999). Informática y teorías del aprendizaje. Píxel-Bit. *Revista de medios y educación, 12*, 87-100.

Velázquez-Iturbide, J. Á. (2018). *Report of the Spanish Computing Scientific Society on Computing Education in Pre-University Stages*. Paper presented at the TEEM'18: Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality, Salamanca, 2018.

Walsh, B., & Dolan, R. (2009). *A guide to teaching practice in Ireland*: Gill & Macmillan Ltd.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology, 25*(1), 127-147. https://doi.org/10.1007/s10956-015-9581-5

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

Werner, L., Denner, J., Campe, S., & Kawamoto, D. C. (2012). *The fairy performance assessment: measuring computational thinking in middle school.* Paper presented at the 43rd ACM technical symposium on Computer Science Education.

Wing, J. M. (2006). Computational Thinking. *Communications of the ACM, 49*(3).

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 366*(1881), 3717-3725. https://doi.org/10.1098/rsta.2008.0118

Wing, J. M. (2011). *Computational thinking.* Paper presented at the 2011 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC).

Yadav, A., Gretter, S., Good, J., & McLean, T. (2017). Emerging Research, Practice, and Policy on Computational Thinking. In R. P. & H. C. (Eds.), *Emerging Research, Practice, and Policy on Computational Thinking. Educational Communications and Technology: Issues and Innovations* (pp. 205-220): Springer, Cham.

Yadav, A., Gretter, S., Hambrusch, S., & Sands, P. (2016). Expanding computer science education in schools: understanding teacher experiences and challenges. *Computer Science Education, 26*(4), 235-254. https://doi.org/10.1080/08993408.2016.1257418

International Trends in K–12 Computer Science Curricula through Comparative Analysis: Implications for the Primary Curricula

Michiyo Oda[1]

Yoko Noborimoto[2]

Tatsusya Horita[1]

[1]Tohoku University

[2]Tokyo Gakugei University

DOI: 10.21585/ijcses.v4i4.102

Abstract

The purpose of this study was to identify international trends in K–12 computer science curricula in countries that have introduced computer science education. Content analysis method was used to analyse the country-wide curricula of 10 countries which have introduced computer science education at the primary level. The K–12 Computer Science Framework was used as a theoretical frame to analyse the curricula. The results show that most countries begin their curricula with sub concepts of *algorithms*, *program development*, and under *impact of computing*, along with the practice of creating computational artifacts; then, countries expand upon computer science concepts and practices as learners progressed through the higher grades. Further, countries tend to introduce computer science concepts and practices in stages; once concepts and practices are introduced, they continue across multiple grades. Three approaches to implementing computer science education into the country-wide curriculum were found: introducing computer science (a) as an independent subject, (b) within multiple subjects, and/or (c) as a part of transversal competencies or an independent computer science curriculum with a cross-curricular approach. These study findings can contribute to a worldwide effort to introduce computer science education at the primary level.

Keywords: Computer science education; computer science curricula; primary school; computational thinking; cross-country analysis.

1. Introduction

1.1 Background

In this increasingly digital world, peoples' lives are heavily influenced by computing (French Academy of Sciences, 2013). This trend has affected people's lives, resulting in structural changes in society and triggering a growing need for a skilled science, technology, engineering, and math (STEM) workforce. For example, Fayer et al. (2017) reported that more than 800,000 STEM jobs were added to the U.S. economy between May 2009 and May 2015, and that STEM occupations grew by 10.5 percent during the same period, which is double the number of non-STEM occupations in the U.S. More importantly, for young people to thrive in this quickly changing, technology-based society, they must understand how computers work and be able to create innovative solutions while collaborating with others.

This societal change has stimulated concern regarding computer science education in K–12 education among European and American industries, academia, and policymakers. Published reports have detailed the need for education reform, including integrating rigorous computer science education rather than information and communication technology (ICT) applications (Association for Computing Machinery, 2014; Computing at School Working Group, 2009; Gander et al., 2013; The Royal Society, 2012). Corresponding with these reports, curriculum reform initiatives have been introduced in various countries in K–12 settings. For example, European Schoolnet (2015) reported that 16 out of 21 European countries participating in their survey have already integrated computer science into their school curricula. Some countries have begun focusing their computer science education efforts within K–12 education, including the primary level. As of 2016, Bocconi et al. reported that policy initiatives of integrating computer science education in primary schools had been in place in several countries in Europe, Oceania, and Asia.

Computer science is a rigorous academic discipline on equal footing with other academic disciplines such as mathematics, physics, or geography, covering principles such as algorithms, data structures, programming, systems architecture, design, and problem solving (Comer et al., 1989; ECDL Foundation, 2015; The Royal Society, 2012). The idea that computer science is a discipline of its own is supported by the notions that computer science includes a body of knowledge that includes widely applicable ideas and concepts, theoretical frameworks to which these ideas and concepts apply, and fundamental concepts that do not change rapidly over time (The Royal Society, 2012). Based on the premise that computer science is an academic discipline, and that young learners should have the opportunity to learn concepts and principles of computer science, the Computing at School Working Group insists that there is a need for a consistent curriculum and body of knowledge on computer science education beginning at the primary level (Computing at School Working Group, 2009). Computer science is the common term in the U.S., while informatics is the common term in Europe (Gander et al., 2013). Although computer science, informatics, and computing have slightly different definitions, they all have been recognized as having a similar meaning (Heintz et al., 2016; Rolandsson & Skogh, 2014). Furthermore, computer science and programming (i.e. coding) are often recognized as equivalent in K–12 education (Association for Computing Machinery et al., 2016). In this study, computer science is used as an umbrella term in the general discussion. We chose the term used in the countries' curricula when discussing the specific countries.

In addition, programs for students enrolled in secondary school through university generally expand upon on the learning that occurs during primary programs. However, when considering computer science education, no successful models of curricula exist yet at the primary level. The introduction of computer science education to this age group is relatively new, as computer science education has been primarily set aside for higher education level learning (Heintz et al., 2016). Therefore, research on computer science curriculum at the primary level is a gap in the literature and must be further investigated. One way to address this issue is to explore international trends in K–12 computer science curricula from countries that have already integrated computer science into its national curricula. Through this exploration, meaningful suggestions may emerge and contribute to the development of primary level curricula.

1.2 Comparative Education Research on K–12 Computer Science Education

There are three approaches to consider in comparative education research on computer science education. The first approach is scrutiny and analysis of the educational system of computer science within one or more countries. For instance, in 2014 and 2015, ACM Transactions on Computing Education (TOCE) published special issues on computer science education in K–12 schools. These special issues included case studies on K–12 computer science education and showed great diversity in computer science education approaches adopted in each country based on each country's history and tradition of the educational system (Tenenberg & McCartney, 2014). In this special issue, Gal-Ezer & Stephenson (2014) compared K–12 computer science education in Israel and the U.S., focused on curriculum development, and reported common issues and challenges in areas in curriculum development and teacher training. Choi et al. (2015) reported current issues in introducing computer science education in K–12 education in Korea by analyzing education environments, including educational systems, curricula, and teaching environments. This first approach shows the detail of each country's computer education status but does not tell the trends of computer science education.

The second approach is comparative research through a cross-country comparative analysis. For example, Bocconi et al. (2016) reported a recent trend in computer science education in compulsory education among European countries through desk research, a survey of the Ministries of Education in 18 countries, and expert semi-structured interviews. One of their findings was that in 10 countries in Europe, including England, Poland, and Italy, computational thinking (i.e. computer science) is included in school curricula at the primary level. Heintz et al. (2016) reviewed the introduction model of K–12 computer science education in 10 countries by analyzing the relevant documents for each country and found that the common model is to make computer science education compulsory in primary school and elective in secondary school. So et al. (2020) introduced computational thinking education (computer science education) status in the Asian Pacific Region by reviewing five empirical studies and one literature review study and showed the current research trends in the field of computer science education and teachers' perception in the region. The second approach presents computer science education trends in worldwide or regions; however, little research has been conducted in terms of curriculum trend.

The third approach is comparative research across multiple countries' reports. For instance, the Ministry of Education, Culture, Sports, Science and Technology (2015) in Japan investigated 23 countries that integrated computer science education into compulsory education from all over the world, including England, Estonia, Germany, Russia, Argentina, Taiwan, and South Africa. These countries were selected using the criteria of higher-ranking countries in PISA 2012 and TIMSS 2011; all countries had implemented computer science education in compulsory education. The ministry conducted desk research and expert interviews on computer science education and published a document combining the countries' status reports. The status of computer science's introduction in primary education was included as a part of each country's report. The third approach also demonstrates computer science education trends worldwide. This approach combines with the first and second approach. However, comparative research on computer science curriculum at the primary education level is still limited.

These three approaches of comparative education research focus primarily on comparing computer science education implementation status among countries and regions. To date, there are only a few studies conducting a cross-country comparative analysis on K–12 curriculum in terms of computer science. In addition, there are only a few studies focusing on primary computer science education in comparative education research. For these reasons, it is imperative that a cross-country comparative analysis be conducted on K–12 computer science curriculum, which will offer implications for primary computer science curriculum and promote consistency in curriculum development nationwide.

1.3 Research Purpose

This study targeted 10 countries that had already introduced computer science education in K-12. The purpose of the research is to show international trends in K–12 computer science curricula, and as a result, get meaningful findings to apply to the primary level curricula. Since the introduction of computer science education at the primary level is relatively new, this study may contribute to the countries' efforts to introduce computer science education at this level. This research is guided by the following research questions:

RQ1. What are the common approaches for computer science education in K–12 among the countries?

RQ2. What are the common trends on K–12 computer science curricula among the countries?

With various definitions of school levels across countries, this paper will use the term K–12 to refer to primary and secondary education. The term primary refers to primary school, which often includes elementary school (except in Sweden, where primary school refers to Years 1–3, middle school refers to Years 4–6, and secondary school refers to Years 7–9). The term secondary refers to both lower secondary and upper secondary; lower secondary includes middle school, junior high school, junior secondary school, lower secondary school, and secondary school; upper secondary refers to high school, senior high school, senior secondary school, upper secondary school, and secondary school.

1.4 Analysis perspective

To analyze the computer science curricula, understanding the perspective of a theoretically founded high-quality curriculum is critical. According to Cuban (1992), the definition of curriculum is "a series of planned events intended for students to learn particular knowledge, skills, and values and organized to be carried out by administrators and teachers" (p. 221). Curriculum is important since it gives direction to instruction in the educational system, and the accumulation of the learning experiences stimulates students' lives and eventually contributes to the quality of their lives (Schmidt et al., 1997).

Schmidt et al. (2005) believed that coherence and rigor are important characteristics that define a high-quality curriculum. According to the researchers, a curriculum is coherent when the sequence of topics and performances appear based on the hierarchical nature of the subject discipline, and the depth of the discipline aligning to the coherence is defined as rigor. They examined the content standards analysis of the TIMSS's top six countries in mathematics and top four countries in science, and they reported that the coherence and rigor of these TIMSS top-achieving countries differed strikingly from the U.S. national standards in mathematics and science. The curricula of TIMSS's top-achieving countries shows a pattern in which new topics are gradually introduced and taught for a few grades, and then different topics are kept in the curriculum. However, various U.S. national standards in mathematics and science show that many more topics are introduced in the first grade and stay longer in each grade than those of the high-achieving countries' curriculum.

In alignment with coherence and rigor, Goodland and Su (1992) stated that organization of the curriculum is critical, which involves scope, continuity, sequence, and integration. Ediger (1995) and Maker (1986) noted that that the high-quality curriculum is designed with scope and sequence. Scope represents the breadth of the curriculum as a horizontal perspective, and continuity and sequence are organized vertically (Goodland & Su, 1992). The scope of the curriculum refers to what is taught, while the sequence is the order and depth of the content tying to the succession of human development (Ediger, 1995; Goodland & Su, 1992). Continuity is the organization of the curriculum through which students revisit the content, and integration aims to involve elements such as concepts, skills, and values in the curriculum (Goodland & Su, 1992). Since continuity is a part of sequence, and integration is a process of curriculum development, this study focuses on scope and sequence as a perspective for curriculum analysis to evaluate a high-quality computer science curriculum.

2. Method

Content analysis method was used to analyze the 10 countries which will be described in great detail in the Content Analysis section.

2.1 Selection of National Curricula Documents

First, we reviewed papers and reports that examined large-scale comparative research on K–12 computer science education status worldwide and within specific regions; the three papers/reports investigated were "Developing Computational Thinking in Compulsory Education" (Bocconi et al., 2016), "Computing our future" (European Schoolnet, 2015), and "Research on Programming Education in International Countries" (Ministry of Education, Culture, Sports, Science and Technology, 2015). Then we identified countries referenced in the papers and reports that matched the following criteria during our investigation (i.e., May 2019 to September 2019):

- Countries that offered computer science curricula or curricula that had integrated computer science in K–12.

- Countries that had implemented country-wide computer science curricula in schools. We excluded countries that had developed curricula at the regional level only, such as Spain, Germany, Belgium, and Switzerland, based on the research by Bocconi et al. (2016).

The reason for setting the criteria as such is to gain the implications for computer science curriculum at the primary level from the perspective of consistency of K–12 curricula; therefore, the countries that have integrated computer science education in K–12 were chosen for this study.

To determine whether the countries referenced in the paper or report matched the above criteria, we conducted additional research using website information, documents published by the ministries of education or relevant institutions, and peer-reviewed papers and reports that review national computer science education integration status. When there were unclear items related to educational systems and computer science education in each country, we asked for clarification from either ministry of education or national researchers who wrote papers to introduce their countries' introduction status of K–12 computer science education. As a result, 10 countries—Australia, England (United Kingdom), Finland, France, Hong Kong, Korea, New Zealand, Poland, Portugal, and Sweden—were selected.

2.2 Collection of National Curricula Documents

The national curricula collection was conducted between May and September 2019. After selecting the 10 countries, we identified the grades and subjects in which the countries introduced computer science education using the three papers and reports mentioned in section 2.1, website information, documents published by the ministries of education or relevant institutions, peer-reviewed papers and reports that reviewed national computer science education integration status, and information from ministries of education and national researchers in the K–12 computer science education field. Then, the 10 countries' K–12 national curricula for the grades and subjects in which the countries introduced computer science education were collected from the website of ministries of education or equivalent institutions. The primary sources for the curricula appear in **Appendix A**. We collected curricula for both compulsory education and post-compulsory education (the first year of primary school through the last year of secondary school). Although most countries have introduced pre-primary education and higher education, and some countries have introduced pre-primary education as compulsory, this study did not include either one. In addition, this study investigated general high school curriculum and did not include other types of high schools, such as technical high schools and vocational schools. Moreover, because several countries have introduced computer science as an elective subject in secondary school, this study involved both compulsory and elective subjects for analysis.

In Sweden, computer science has been offered as a part of digital competence education, and digital competence has been integrated into various subjects in upper secondary school (Skolverket, 2017); therefore, no standalone compulsory courses in computer science education exist. We referred to the commentary by the Swedish National Agency for Education, entitled "Få syn på digitaliseringen på gymnasial nivå" (Skolverket, 2017) to understand the curricula in this country. This is a supplementary material for teachers providing background understanding on the role of digitization in the curriculum (Bocconi et al., 2018). Within this supplementary document, we identified all subjects that were related to digital competence and investigated the identified subjects' curriculum to assess whether the learning contents matched the K–12 Computer Science Framework definitions of the concepts and practices following the process shown in section 2.3. When there were learning contents in the supplemental material that matched the definitions, we decided that the subject was integrated computer science.

2.3 Selection of the Analysis Framework

To identify international trends in K–12 computer science curricula, we conducted a cross-country comparative curriculum analysis. To do so, a theoretical framework was necessary. In this research, the K–12 Computer Science Framework (Association for Computing Machinery et al., 2016) was utilized to analyze the 10 countries' national curricula. The framework outlines key concepts and practices that students should obtain by completing computer science education and was founded on profound research and practices; it was developed by the K–12 Computer Science Framework Steering Committee, which includes the Association for Computing Machinery, Code.org, Computer Science Teachers Association, Cyber Innovation Center, and National Math and Science. The framework is organized into five core concepts, which are divided into 17 subconcepts (**Appendix B**), and seven core practices, which are divided into 23 achievement levels by the end of Grade 12 (**Appendix C**).

The Association for Computing Machinery et al. (2016) recommended that the concepts and practices should be implemented into the school curriculum to provide a meaningful computer science experience for students. According to the organization, the structure of the framework was intended to align with the structure of widely accepted education frameworks, such as the framework for K–12 Science Education (National Research Council, 2012). According to the Association for Computing Machinery et al. (2016), the core concepts "represent major content areas in the field of computer science", and the practices "are the behaviors that computationally literate students use to fully engage with the core concepts of computer science" (p. 52). The concepts and practices in the framework hold main content areas and behaviors to implement the computer science curriculum for all students in K–12 education.

Although the framework was developed in the U.S., we utilized it to conduct a cross-country curriculum analysis because the framework (a) benchmarks several countries outside of the U.S., including the United Kingdom, Germany, Poland, and New Zealand, introducing a multi-country perspective; (b) is based on rigorous research and builds on some of the oldest K–12 computer science curriculum; and (c) represents a "baseline literacy for all students" without providing advanced contents to study (Association for Computing Machinery et al., p. 15).

2.4 Content Analysis

Content analysis method was employed in this study. According to Cohen, Manion, and Morrison (2007), the content analysis process includes defining words or sentences in texts, coding and categorizing them, and counting the words, codes, and categories. This study utilized the content analysis process that Cohen, Manion, and Morrison (2007) provide for curriculum analysis.

2.4.1 Analysis Contents Identification

First, we identified the chapters or units that described the knowledge and skills that are expected to be taught in each of the 10 countries' curricula. Madaus and Kellaghan (1992) explained that curriculum includes six major components: (a) context, (b) broad educational aims, (c) objectives of specific curricula or learning units, (d) curricular materials, (e) transactions and process, and (f) outcomes. In this study, we investigated representing the third component, objectives of specific curricula or learning units, which includes knowledge and skills to be taught that provides a basis for designing classroom instructional activities (Madaus & Kellaghan, 1992). A diverse range of chapter or unit titles were identified in the 10 countries' curricula related to computer science education objectives and learning units; these titles differed by country, subjects, and grades. For example, Australia utilizes the title "Sequence of content F-10" in the digital technologies subject, and England uses the title "subject content" in the computing subject. A list of the chapter or unit titles in curricula included in the content analysis appears in **Appendix D**. For example, we identified the "progress outcome" within the two technological areas of "computational thinking for digital technologies" and "designing and developing digital outcomes" as the above mentioned third component from the New Zealand, Technology curriculum.

2.4.2 Division of Sentences

Each curricula was composed of multiple sentences. We divided these multiple sentences, identified in section 2.4.1, into single sentences. For example, in the New Zealand curriculum exemplified in the section 2.4.1, multiple sentences were included in the "progress outcome" within the technological areas of "computational thinking for digital technologies" and "designing and developing digital outcomes". We divided the sentences which were included in the "progress outcome" into single sentences. An example of a divided single sentence was "In authentic contexts and taking account of end users, students use their decomposition skills to break down simple non-computerised tasks into precise, unambiguous, step-by-step instructions (algorithmic thinking)" (Progress outcome 1). When the texts were not written in English and we could not find an English version, Google Translate was utilized to translate the text from the original language to English to understand the meaning. Li et al. (2014) investigated the reliability of using Google Translate by comparing Google Translate with human translation, finding that Google English translation showed a high correlation with both human English translation and an original Chinese text. Although this research investigated Chinese-to-English translation, we thought this result could be applied to other languages. When there were unclear words in the curriculum that could not be translated using Google Translate, we asked researchers in the field of K–12 computer science education from the country so that we could understand the meaning.

2.4.3 Separation of Sentences

After dividing the multiple sentences from the curricula into single sentences, we divided the sentences into smaller clauses with identifiable meanings. For example, in the New Zealand curriculum exemplified in the section 2.4.2, the sentence, "In authentic contexts and taking account of end users, students use their decomposition skills to break down simple non-computerised tasks into precise, unambiguous, step-by-step instructions (algorithmic thinking)" (Progress outcome 1) was further divided into two clauses; "In authentic contexts and taking account of end users" and "students use their decomposition skills to break down simple non-computerised tasks into precise, unambiguous, step-by-step instructions (algorithmic thinking)".

2.4.4 Tagging of Units

After dividing each single curriculum sentence into one or more clauses, we tagged each clause with the concepts (5 core concepts and 17 subconcepts) and the practices (seven practices and 23 achievement levels of the practices) based on the definition from the K–12 Computer Science Framework. When the clauses were categorized across multiple concepts or practices, we tagged them all. When the clauses did not have an appropriate concept or practice to categorize, we did not tag them, and removed them from further analysis. For example, in the New Zealand curriculum example, "In authentic contexts and taking account of end users" was tagged with practice 1.2. The sentence "students use their decomposition skills to break down simple non-computerised tasks into precise, unambiguous, step-by-step instructions (algorithmic thinking)" was tagged with both *algorithms* and practice 3.2. The definition of practice 1.2 is "students should recognize that users of technology have different needs and preferences and that not everyone chooses to use, or is able to use, the same technology products" (Association for Computing Machinery et al., 2016, p. 74). The definition of *algorithms* is "Algorithms are designed to be carried out by both humans and computers. In early grades, students learn about age-appropriate algorithms from the real world. As they progress, students learn about the development, combination, and decomposition of algorithms, as well as the evaluation of competing algorithms" (Association for Computing Machinery et al., 2016, p. 91). In addition, practice 3.2. represents decomposing real-world problems into manageable subproblems (Association for Computing Machinery et al., 2016). As shown in this example, we compared each clause to the definition of the concepts and practices from the K–12 Computer Science Framework. When the definition matched the meaning of the clauses, the clauses were tagged with the concepts and practices.

2.4.5 Review

One researcher categorized the clauses according to the concepts and practices, and the other researchers, who are subject specialists in computer science education, reviewed them. All researchers jointly discussed the categorization when there was a discrepancy to reach an agreement. We used this content analysis approach to generate country mapping tables by the concepts and practices (Tables 5 and 6).

3. Results

3.1 RQ1. What are the common approaches for computer science education in K–12 among the countries?

Table 1 summarizes the educational system and state of computer science education introduction by country. All countries revised their national curricula within a few years of the first implementation and introduced or reinforced computer science education. The country with the earliest development of a computer science K–12 curriculum was England in 2014, followed by Australia in 2015, Finland in 2016, and then France in 2016.

Table 2 displays the computer science integration curricula by country. Although the duration of compulsory education varies among countries, all of the countries introduced computer science as compulsory subjects during compulsory education. Some countries changed computer science from a compulsory subject to an elective subject as grade level increases. In addition, most of the countries in this study, except for Korea, integrated computer science curricula into a single subject, or multiple similar subjects, throughout compulsory education. Alternatively, Korea integrated computer science into "Practical Art" at the primary level, and then offers informatics at the secondary level.

Table 3 provides computer science integration subjects. "Technology" represents the subjects that include technology in the subject name, including Technology, Digital Technologies, General Study (Technology), Practical Arts (Technology/Home Economics), and Digital Science and Technology. "Independent computer science subject" stands for the independent computer science subjects, including Computing, Informatics, ICT, Digital and Computer Science, and Computer Applications B. Technology was the most introduced subject, followed by the independent computer science subject. Subjects of crafts, social sciences, language, and transversal competencies were found in only one country.

Table 4 demonstrates computer science introduction approaches. Several approaches for introducing computer science education were found. The first approach was introducing computer science as independent subjects (Australia, England, France, Hong Kong, Korea, New Zealand, Poland, and Portugal). The second approach was integrating computer science within multiple subjects (Finland, France, and Sweden). The third approach was introducing computer science as a part of transversal competencies or an independent computer science curriculum with a cross-curricular approach (Finland and Portugal). Some countries have introduced mixed approaches, differing by grade level. For example, Portugal introduced the third approach during lower primary level and transitioned to the first approach beginning in the upper primary level. In addition, some countries introduced different subjects within the same approach. For example, Portugal introduced computer science within ICT during grades 1–9 and changed the subject to Computer Applications B in grade 12.

Table 1. Educational System and State of Computer Science Education Introduction

Countries	Educational system and state of computer science education integration
Australia	In Australia, school education is compulsory between the ages of six and 16 (Year 1 to Year 9 or 10) (Australian Trade and Investment Commission, n.d.). Computer science education starts during the first year of school and continues through Year 10; there is no mandated national curriculum at the final stage of secondary school (Year 11 and Year 12) (Falkner et al., 2019). Current Foundation –Year 10 Australian Curriculum has been in place since **2015**, and the next revised version will be published by the start of 2022 (Australian Government, Department of Education Skills and Employment, 2020).
England	Compulsory education in England includes schooling for children age five through 16 (Years 1–11) (GOV.UK, n.d.). At the years 10 and 11, students can additionally take GCSE (Falkner et al., 2019). Computer science is integrated into Computing

	curriculum, which was replaced by ICT curriculum in **2014** (Computing at School, 2013).
Finland	In Finland, compulsory education is between the ages of seven and 15 (Grades 1–9) and compulsory pre-primary education starts at the age of six. In the revised curriculum for basic education, computer science is integrated in mathematics, crafts, and transversal competencies. This new curriculum was implemented for grades 1–6 in **2016**. For grades 7–9, the implementation of the revised curriculum was staged in **2017, 2018, and 2019** (National Board of Education, n.d.).
France	Education is compulsory in France from age six to 16 (Grades 1–11) (Gueudet et al., 2017). There are three types of schools in upper secondary level: general, technological, and vocational (Gueudet et al., 2017). This study used curricula under the general school level for analysis. The revised curriculum of the primary and the lower secondary schools (Grades 1–9) was implemented in September **2016** (Gueudet et al., 2017). For upper secondary school, the course "Digital Science and Technology" became compulsory for the second class of upper secondary school (Grade 10) in **2019**, and "Digital and Computer Science" has been provided as an elective in the first class of upper secondary school (Grade 11) (Ministry of National Education and Youth, 2018).
Hong Kong	In Hong Kong, compulsory education is between the ages of six and 15 (Primary 1–6, Secondary 1–3). The revised curricula were implemented in **2017** (The Curriculum Development Council, 2017a, 2017c, 2017b). Computer science education is integrated into technology education from Primary 1 to Secondary 6. Technology education is subsumed in the General Studies curriculum at the primary level. At the lower secondary level, schools are adopting a subject-based learning approach. Computer science is offered within the "Computer Literacy" subject to implement learning element modules of the technology education curriculum. At the upper secondary level, the elective subject "Information and Communication Technology (ICT)" has been offered as technology education (The Curriculum Development Council, 2017c).
Korea	In Korea, compulsory education is between ages six and 15 (Primary 1–6, Secondary 1–3) (National Center on Education and the Economy, n.d.). The revised elementary school curriculum for 5th and 6th graders was implemented in **2019** (Korean Ministry of Education, 2015). The previous curriculum included an ICT unit in "Practical Art" for 12 teaching hours, but the new curriculum includes computer science in "Practical Art" for more than 17 teaching hours (The Institute for Democracy, 2018). For lower and upper secondary schools, the introduction of the revised curriculum was staged in **2018, 2019, and 2020**, depending on grades (Korean Ministry of Education, 2015). The lower secondary school informatics course was an elective in previous curriculum, but

	is now a compulsory course (The Institute for Democracy, 2018). The new upper secondary informatics course is elective, consistent with the prior curriculum (The Institute for Democracy, 2018).
New Zealand	Compulsory education in New Zealand is between the ages of six and 16 (Years 1–10). The revised technology curriculum (Years 1–13) was published in **2017** and is expected to be fully implemented by the start of the 2020 school year (Ministry of Education, 2017). The new curriculum strengthened digital technologies, including computational thinking for digital technologies and designing and developing digital outcomes, as a part of the curriculum (New Zealand Ministry of Education, n.d.).
Poland	In Poland, compulsory education starts at the age of seven and lasts until the completion of Year 8 in primary school (Grades 1–8) (European Commission, 2019). The revised informatics curriculum was implemented in primary school (Grades 1–8) in **2017** (Polish Republic, 2017). Post-primary schools include 4–year general secondary school and 5–year technical secondary school. For secondary education (Grades 9–12, 4–year general secondary school, or Grades 9–13, 5–year technical secondary school), the revised curricula were applied in the **2019/2020** school year in the first grade (Education Development Center & Ministry of Education, 2019). Informatics is mandatory for one hour per week for three years; an elective extended informatics course is available for two to three hours per week for three years during secondary education in the new curriculum.
Portugal	In Portugal, compulsory education lasts for 12 years, starting at age six and ending at age 18 (European Commission, 2017a). Compulsory education includes basic education, which is between ages six and 15 (Grades 1–9), and secondary education, which is between ages 15 and 18 (Grades 10–12) (European Commission, 2017a). The revised ICT curriculum was implemented in basic education (Grades 1–9) and Computer Applications B (Grade 12) in secondary education in the **2018/2019** school year (Directorate-General for Education & Government of the Portuguese Republic, 2018).
Sweden	In Sweden, compulsory school is between the ages of seven and 16 (Years 1–9) (European Commission, 2017b). The revised curricula was implemented in basic education in **2017** and became compulsory in **2018** for all schools (Bocconi et al., 2018). Upper secondary school (Years 10–12) is optional. A revised version of the upper secondary school curriculum was published in 2018 for strengthening students' digital skills.

Table 2. Computer Science Integration Curricula

Countries	Compulsory Education (Starting in the first year of primary school)	Post-Compulsory Education (Ending in the last year of secondary school)
Australia	Digital Technologies (F–8), Digital Technologies (Grades 9–10) *	-
England	Computing (Years 1–11)	-
Finland	Mathematics (Grades 1–9), Crafts (Grades 3–9), Transversal competencies (ICT competences) (Grades 1–9)	-
France	Mathematics (Grades 1–10), Mathematics (Grade 11) *, Science and Technology (Grades 4–6), Technology (Grades 7–9), Digital Science and Technology (Grades 10), Digital and Computer Science (Grade 11) *	-
Hong Kong	General Study (Primary 1–6), Technology (Lower Secondary 1–3)	Technology (Upper Secondary1–3) *
Korea	Practical Arts (Technology/ Home Economics) (Primary 5, 6), Informatics (Lower secondary 1–3)	Informatics (Upper secondary 1–3) *
New Zealand	Technology (Years 1–10)	Technology (Years 11–13) *
Poland	Informatics (Grades 1–8)	Informatics (Grades 9–11), Informatics (Grades 9–11) *
Portugal	ICT (cross curricular in Grades 1–4, Grades 5–9), Computer Applications B (Grade 12) *	
Sweden	Mathematics (Grades 1–9), Technology (Grades 1–9), Social Studies (Grades 4–9)	Social Studies (Grades 10–12), Swedish / Swedish as a second language (Grades 10–12), Computers and ICT (Grades 10–12) *, Information and communication (Grades 10–12) *, Technology (Grades 10–12) *

Note. * Signifies an elective subject.

Table 3. Computer Science Integration Subjects

Approaches	AL	EN	FN	FR	HK	KR	NZ	PL	PR	SW
Technology	✓*			✓	✓*	✓	✓*			✓
Independent computer science subjects		✓		*		✓*		✓*	✓*	*
Mathematics			✓	✓*						✓
Crafts			✓							
Social Studies										✓
Language										✓
Transversal competencies			✓							

Note. ✓ signifies a compulsory subject; * signifies an elective subject. AL (Australia), EN (England), FN (Finland), FR (France), HK (Hong Kong), KR (Korea), NZ (New Zealand), PL (Poland), PR (Portugal), SW (Sweden).

Table 4. Computer Science Introduction Approach

Approaches	AL	EN	FN	FR	HK	KR	NZ	PL	PR	SW
Independent subjects	✓*	✓		*	✓*	✓*	✓*	✓*	✓*	
Multiple subjects			✓	✓*						✓*
Transversal competencies or independent curriculum with a cross-curricular approach			✓						✓	

Note. ✓ signifies a compulsory subject; * signifies an elective subject. AL (Australia), EN (England), FN (Finland), FR (France), HK (Hong Kong), KR (Korea), NZ (New Zealand), PL (Poland), PR (Portugal), SW (Sweden).

In summary, these results show that all 10 countries introduced or reinforced computer science education during compulsory education, as compulsory subjects, as a result of recent curriculum reform. It was common for countries to introduce computer science as a single subject or multiple similar subjects throughout compulsory education. The approaches that countries used to introduce computer science curriculum included (a) in independent subjects, (b) within multiple subjects, and (c) as a part of transversal competencies or an independent computer science curriculum with a cross-curricular approach. Subjects within the computer science curriculum most often included technology, followed by independent computer science subjects and mathematics.

3.2 RQ2. What are the common trends on K–12 computer science curricula among the countries?

This study analysed 10 countries' computer science curricula for common trends in concepts and practices. Tables 5 and 6 offer a description of each country's inclusion of specific concepts and practices using circle pie charts. With respect to the right half of the circle, the top third of the semi-circle represents grades 1 and 2, the next third represents grades 3 and 4, the last third represents grades 5 and 6. On the left side of the circle, the bottom half of the semi-circle represents grades 7–9, and the next half represents grades 10 through 11, 12, or 13, depending on each country's educational system. In the circles, the colour black represents compulsory subjects, while grey represents elective subjects. When the computer science subjects existed within the curricula, but the concepts or practices did not match, the circle remained blank (white). If there were no computer science subjects, the circle was not drawn (without border). In addition, when a concept or practice was included in at least one or more grades within a specific fan shape that represents several grades, the fan shape was shaded black or grey. For example, if a concept or practice was included in grade 3 but not included in grade 4, the fan shape of grades 3 and 4 was filled in black or grey, even if the concept included in grade 3 and not included in grade 4. This rule was applied to other grades.

3.2.1 Concepts Analysis in Terms of Scope

As shown in Table 5, most countries covered most concepts. On the other hand, in Finland, the sub concepts of *computing systems* and *networks and the internet* were scarcely covered. In addition, several sub concepts, such as *variables*, *control*, and *modularity* in *algorithms and programming* were lacking across countries. Finland is the only country that introduced computer science within multiple subjects, excluding the subject of technology. Other countries that integrated computer science within multiple subjects included technology as one of subjects.

The results indicate that most countries in this study include almost all concepts from the K–12 Computer Science Framework in their computer science curricula. The countries that approached computer science integration with multiple subjects but did not include technology or independent computer science within the subjects (e.g., Finland), had a tendency for excluding *computing systems* and *networks and the internet* in their curricula.

3.2.2 Practice Analysis in Terms of Scope

Table 6 presents the description status in each country by practice. According to the Association for Computing Machinery et al. (2016), both the concepts and practices should be introduced to provide meaningful computer science experiences for students rather than only focusing on concepts alone. Some of the countries covered the concepts well, but lacked reference to practices 1 and 2. Practice 1 represents "considering the needs of diverse users during the design process is essential to producing inclusive computational products" (Association for Computing Machinery et al., 2016, p. 74) and practice 2 represents "the process of performing a computational task by working in pairs and on teams" (Association for Computing Machinery et al., 2016, p. 75). England lacked both practice 1 and 2, followed by Korea and Poland, which lacked practice 1, and Sweden, which lacked practice 2.

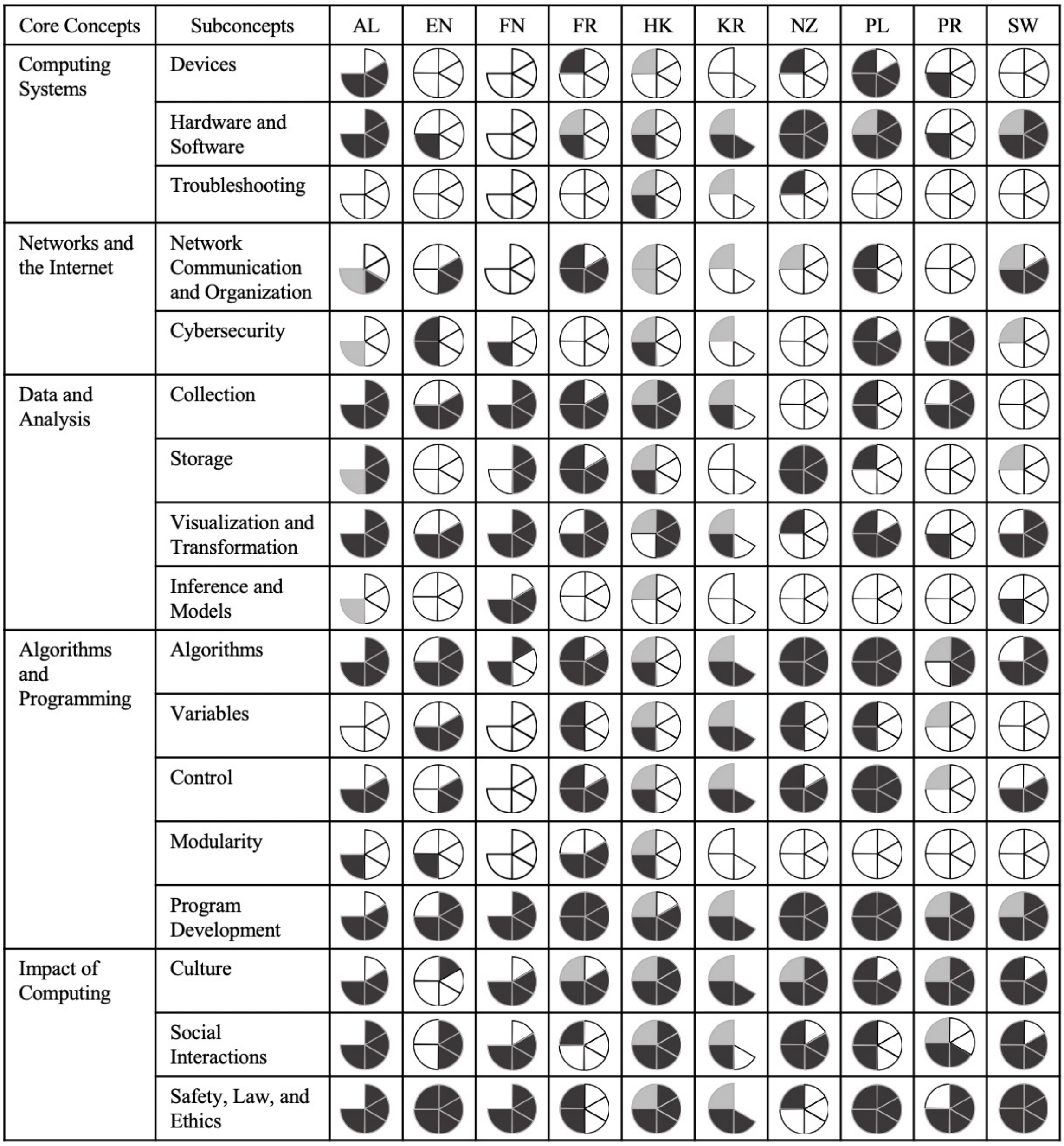
Practices 3 to 6 refer to delineated computational thinking in K–12 Computer Science Framework (Association for Computing Machinery et al., 2016). Computational thinking is commonly mentioned in relation to computer science. Computational thinking is defined as “the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent” (Wing, 2010, p. 1). Computational thinking is a thinking process to maximize the power of computing to solve problems effectively with the thinking process prior to the coding. According to Association for Computing Machinery et al. (2016), computational thinking is at the heart of the computer science practices. On the other hand, several papers argue that computational thinking is more than that. Denning (2017) compared traditional computational thinking with new computational thinking that are generated by Jeannette Wing’ influential statement on computational thinking that was published in 2006 and the resulting discussions on computational thinking and argues that programming ability nurtures computational thinking in traditional computational thinking but learning certain concepts nurtures programming ability in new computational thinking. In line with the argument by Denning, Y. Li et al. (2020) insist that computational thinking should not be restricted in computer science; rather, it is pervasive in daily lives and occupations.

Australia, England, France, Hong Kong, New Zealand, Poland, and Portugal described practices 3 to 6 in their curricula across multiple grades. These countries have asserted computational thinking’s importance in their publications by their respective ministries of education or equivalent organizations (Australian Curriculum, Assessment, and Reporting Authority, 2015; Baron et al., 2014; Department for Education, 2013a; Ministry of Education, 2017). Some countries lacked just one practice among the four practices. For example, Sweden lacked practice 3 and Korea lacked practice 6. Practice 3 represents recognizing and defining computational practices, while practice 6 stands for testing and refinement of computing artifacts. Adding these missing practices into curricula may reinforce nurturing computational thinking in these countries.

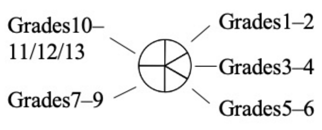
3.2.3 Concepts Analysis in Terms of Sequence

As can be seen in Table 5, common trends were found in the concepts presented in curricula across all 10 countries. Countries had similar tendencies in their timing of introducing certain concepts. For instance, *algorithms, programming,* and the sub concepts under *impact of computing* were described in curriculum in all countries, with descriptions starting in lower primary level and continuing throughout upper secondary level. Seven out of nine countries introduced *algorithms, program development,* and *safety, law, and ethics* at grades 1–2. These subjects continued throughout to the secondary level. Other sub concepts under *computing systems* and *networks and the internet* demonstrated a different trend from the *algorithms, programming,* and the sub concepts under *impact of computing*. The concepts had a tendency to be described from upper grades. For example, nine out of 10 countries described *hardware and software* and six out of 10 countries described *cybersecurity* at grades 7–9. Similarly, six out of 10 countries described *variables* at grades 7–9, and seven out of 10 countries described *control* at grades 5–6, which are the sub concepts under *algorithms and programming*. These sub concepts continued during multiple grades. It would appear that these sub concepts are to be learned in the higher-grade levels.

Table 5. Country Mapping by Concepts



Core Concepts	Subconcepts	AL	EN	FN	FR	HK	KR	NZ	PL	PR	SW
Computing Systems	Devices										
	Hardware and Software										
	Troubleshooting										
Networks and the Internet	Network Communication and Organization										
	Cybersecurity										
Data and Analysis	Collection										
	Storage										
	Visualization and Transformation										
	Inference and Models										
Algorithms and Programming	Algorithms										
	Variables										
	Control										
	Modularity										
	Program Development										
Impact of Computing	Culture										
	Social Interactions										
	Safety, Law, and Ethics										



● Computer science subjects exist within the curricula, and subconcepts are described within **compulsory subjects** at this grade level.

◍ Computer science subjects exist within the curricula, and subconcepts are described within **elective subjects** at this grade level.

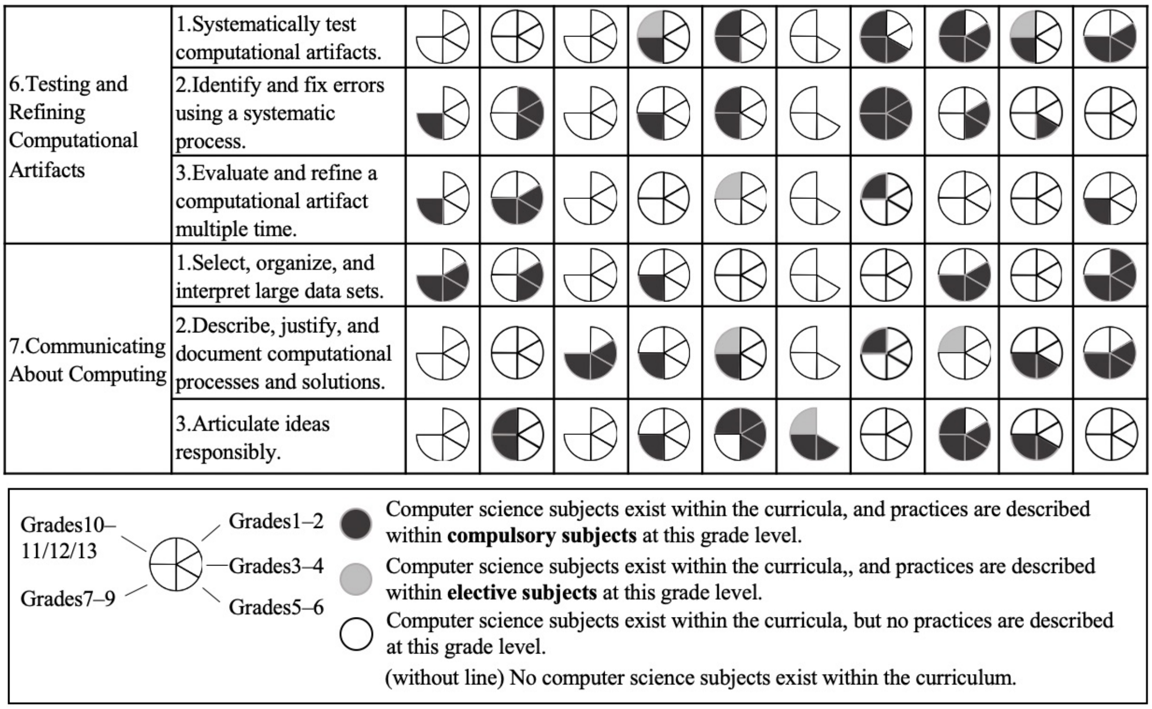
○ Computer science subjects exist within the curricula, but no subconcepts are described at this grade level.

(without line) No computer science subjects exist within the curriculum.

Note. AL (Australia), EN (England), FN (Finland), FR (France), HK (Hong Kong), KR (Korea), NZ (New Zealand), PL (Poland), PR (Portugal), SW (Sweden). Australia: Compulsory education curriculum starts from Foundation, so this study includes Foundation in grades 1–2. The bottom half of the circle of the left side represents grades 7–10. New Zealand and Poland: The bottom half of the circle of the left side represents grades 7–8 and the next half represents grades 9–11/13.

Table 6. Country Mapping by Practices

Core Practices	By the end of Grade 12, students should be able to	AL	EN	FN	FR	HK	KR	NZ	PL	PR	SW
1.Fostering an Inclusive Computing Culture	1.Include the unique perspectives of others.	○	○	○	○	◑	○	○	○	◕	○
	2.Address the needs of diverse end users.	◕	○	◕	◔	◔	○	●	○	○	◕
	3.Employ self- and peer-advocacy.	◑	○	◕	○	◔	○	◔	○	○	○
2.Collaborating Around Computing	1.Cultivate working relationships.	◕	○	◕	◔	○	◔	◔	●	◑	○
	2.Create team norms, expectations, and equitable workloads.	◔	○	◔	◕	◑	○	○	◑	◕	○
	3.Solicit and incorporate feedback.	○	○	○	○	○	○	○	○	○	○
	4.Evaluate and select technological tools.	○	○	○	◔	◔	○	○	◕	◕	○
3.Recognizing and Defining Computational Problems	1.Identify complex, interdisciplinary, real-world problems.	◕	○	○	◔	◑	◑	◔	◕	◑	○
	2.Decompose complex real-world problems.	◔	◑	○	◔	◑	◑	◕	◔	○	○
	3.Evaluate whether it is appropriate and feasible.	○	○	○	○	○	○	○	○	○	○
4.Developing and Using Abstractions	1.Extract common features.	○	◔	○	○	○	◔	◑	◔	◔	○
	2.Evaluate existing technological functionalities.	◔	○	○	◔	○	○	◕	○	◑	○
	3.Create modules and develop points of interaction.	◔	◔	○	○	◔	○	◑	◔	○	○
	4.Model phenomena and processes and simulate systems.	◔	◔	○	◕	○	◔	○	○	○	◕
5.Creating Computational Artifacts	1.Plan the development of a computational artifact.	◕	◕	◑	◕	◑	◕	◕	●	◔	◕
	2.Create a computational artifact.	◕	◕	◕	●	◕	◕	●	●	●	◕
	3.Modify an existing artifact.	◑	○	○	◔	◔	○	○	○	◑	○



6.Testing and Refining Computational Artifacts	1.Systematically test computational artifacts.
	2.Identify and fix errors using a systematic process.
	3.Evaluate and refine a computational artifact multiple time.
7.Communicating About Computing	1.Select, organize, and interpret large data sets.
	2.Describe, justify, and document computational processes and solutions.
	3.Articulate ideas responsibly.

● Computer science subjects exist within the curricula, and practices are described within **compulsory subjects** at this grade level.

● Computer science subjects exist within the curricula,, and practices are described within **elective subjects** at this grade level.

○ Computer science subjects exist within the curricula, but no practices are described at this grade level.

(without line) No computer science subjects exist within the curriculum.

Note. AL (Australia), EN (England), FN (Finland), FR (France), HK (Hong Kong), KR (Korea), NZ (New Zealand), PL (Poland), PR (Portugal), SW (Sweden). Australia: Compulsory education curriculum starts from Foundation, so this study includes Foundation in grades 1–2. The bottom half of the circle of the left side represents grades 7–10. New Zealand and Poland: The bottom half of the circle of the left side represents grades 7–8 and the next half represents grades 9–11/13.

Within the concept of *data and analysis*, eight out of 10 countries included sub concepts of *collection* and all countries included *visualization and transformation*, although other sub concepts in the same core concept, such as *inference and models*, were included in four out of 10 countries. The reason for this result could be that *collection* and *visualization and transformation* were taught through the curriculum for mathematics in Finland, France, and Sweden; therefore, the sub concepts could have been reinforced by mathematics.

Troubleshooting was described the least across all 10 countries. Association for Computing Machinery et al. (2016) defined troubleshooting as follows: “When computing systems do not work as intended, troubleshooting strategies help people solve the problem” (p. 89). This sub concept addresses hardware troubles and solving problems through applying hardware and software knowledge. In this context, some sub concepts under *computing systems*, *networks and the internet*, and *algorithms* and *programming* may have supported learning of the concept *troubleshooting.* Therefore, the sub concept of *troubleshooting* requires higher-order thinking skills and various prerequisite knowledge, such as synthesizing and analysing the problems; therefore, this may be why the number of *troubleshooting* descriptions was less common among the countries.

These results suggest common tendencies among the 10 countries regarding the sequence of computer science concepts presented across grade levels. For example, K–12 computer science curriculum usually starts with the concepts of *algorithms, program developmen*t, and the sub concepts under *impact of computing*, during the lower primary level and continues through to the upper primary level. Although *variables*, *control*, and *modality* are under the same core concepts of *algorithms and programming,* these sub concepts tend to be introduced in upper grades. Similarly, *computing systems* and *network and the internet* tend to be introduced in the upper grades.

Similarly, common trends across the 10 countries included each country described the sub concepts in stages and did not describe all concepts at the same grade. After the sub concepts described, the sub concepts continued during multiple grades. For example, French curriculum described sub concepts in stages such as *program development* in Cycle 2 (Grades 1–3), *algorithms* in Cycle 3 (Grades 4–6), and *variables* in Cycle 4 (Grades 7–9). After the sub concepts are described, these continued during multiple grades. This tendency noted in the French curriculum was similar to tendencies in the other nine countries' curricula. The results of this study are similar to results found by Schmidt et al. (2005). The researchers investigated curricula of the highest-achieving TIMSS countries in mathematics and science to identify elements of high-quality curriculum, finding patterns in how new topics are gradually introduced, continued for several grades, and then transition into different topics in the curriculum. Even though Schmidt et al.'s (2005) research investigated mathematics and science, the high-quality curriculum may show the same tendency. In the present study, we found that practices tended to be described only a few grades or multiple grades without consistency except for practice 5 and 6, unlike the sub concepts.

3.2.4 Practice Analysis in Terms of Sequence

As explained in the previous section, practices 3 to 6 were delineated as computational thinking in the K–12 Computer Science Framework (Association for Computing Machinery et al., 2016). As can be seen in Table 6, the number of countries that described practices 3, 4, and 6 tended to increase from lower secondary level to upper secondary level. On the other hand, practice 5, especially practice 5.2, was mentioned most commonly among all countries' curriculum across lower primary through upper secondary levels. The number of countries mentioning practice 5.1 in their curriculum increased from upper primary level to upper secondary level, and the same trend was seen with practice 6.1.

These results indicate that the introduction of computational thinking does not generally happen in order from practices 3 to 6; rather, it starts from creating computational artifacts as represented in practice 5.2 followed by the practices of planning the development of computational artifacts (practice 5.1) and testing computational artifacts (practice 6). Then moves to recognizing and defining computational problems (practice 3) and developing abstractions (practice 4) in the upper grades.

These results also suggest that computer science curricula tend to start not only with learning the concepts of *algorithms* and *program development*, but also with creating computational artifacts at the lower primary level. In addition, the introduction of computational thinking does not necessarily follow the order of practices 3 to 6; rather, it starts with creating computational artifacts (practice 5) and then expands into other practices gradually. Similar trends in timing of introduction were found between concepts and practices among the 10 countries.

4. Discussion

As the results showed, all 10 countries in this study have introduced or reinforced K–12 computer science education. The approaches that the 10 countries used to introduce computer science curriculum included either (a) as independent subjects, (b) within multiple subjects, and/or (c) as a part of transversal competencies or an independent computer science curriculum with a cross-curricular approach.

High-quality curricula require the existence of both scope and sequence. In addition, a high-quality computer science curricula should integrate both concepts and practices (Association for Computing Machinery et al., 2016); therefore, the scope and sequence of the curricula in this study are discussed in relation to concepts and practices in this section.

In terms of scope, most concepts in K–12 computer science curricula were described by nine countries; however, Finland, which integrated computer science within multiple subjects, but did not include the concept of technology or individual computer science as one of the subjects, also showing a lack in the sub concepts under *computing systems* and *networks and the internet*. Since *computing systems* and *networks and the internet* were commonly found in technology or independent computer science subjects, introducing computer science without technology or independent computer science subjects could cause the lack of some concepts for students. However, this result does not mean that students in Finland do not learn the concepts because of following two reasons: (1) This study analysed the third component, objectives of specific curricula or learning units in curricula, as mentioned in the research methodology part, and does not include other parts of curricula, textbooks, or supplemental materials for the investigations, and (2) compared to other countries' curricula, Finland's curricula for the analysis included less information. In addition, Korea and New Zealand covered *networks and the internet* as an elective subject in upper secondary level. This means some students may not have a chance to learn *networks and the internet*; therefore, these countries could introduce *networks and the internet* as compulsory subjects to reinforce the concepts.

From the analysis of practices, practices 3 to 6 are delineated computational thinking in K–12 Computer Science Framework (Association for Computing Machinery et al., 2016). Seven countries among the 10 described practices 3 to 6 at one or some grade levels in their curricula. The three countries who did not reference practices 3 to 6 could further support students' computational thinking by adding these lacking practices to their curricula.

As for sequence, common trends among the 10 countries regarding the introductory grade levels of certain concepts and practices were identified. For example, *algorithms*, *program development,* and the sub concepts under *impact of computing*, were referenced in lower primary level through upper secondary level curricula. Sub concepts such as *hardware and software*, *cybersecurity*, *variables*, and *control* tended to be referenced in upper grades only. Practical activities, such as creating computational artifacts, were described in lower primary level through upper secondary level curricula. The results also suggest that nurturing students' computational thinking can be achieved by implementing practices 3 to 6, but not necessarily in the original order. Practice 5 is generally introduced in lower grades and then expanded upon through practices 3 and 6 in the upper grades.

The implications for computer science curricula at the primary level from this study are summarized as follows:

1) Computer science concepts start from *algorithms*, *program development*, and the sub concepts under *impact of*

computing at lower primary level, then other concepts such as *computing systems* and *networks and the internet* are introduced in upper grades.

2) Computer science practices beginning with creating computational artifacts (practice 5) at lower secondary level, then expand to recognizing problems (practice 3), developing abstractions (practice 4), and testing and refining computational artifacts (practice 6) in upper grades.

3) Concepts and practices are introduced in stages, and after the concepts and practices are introduced, they continue across multiple grades.

4) The introduction of computational thinking does not happen in order from practices 3 to 6. It starts with practice 5 and expands into practices 3 and 6 in the upper grades.

Schools can translate the above implications into practice by first identifying an approach that fits best within their context, introducing computer science curricula as either (a) independent subjects, (b) within multiple subjects, and/or (c) as a part of transversal competencies or an independent computer science curriculum with a cross-curricular approach. Including technology or independent computer science subjects as one of subject offered is preferable so that the concepts of *computing systems* and *networks and the internet* are to be covered in the curriculum. Second, computer science education should be integrated within a single subject or similar multiple subjects throughout compulsory education. In this way, computer science curricula could remain consistent across grade levels.

5. Conclusion

This study analysed the curricula of 10 countries that have introduced computer science education beginning at the primary level with the goal of identifying trends in K–12 computer science curricula. These results offer meaningful findings for the development of computer science curricula at the primary level. A cross-country comparative curriculum analysis was completed, using the K–12 Computer Science Framework (Association for Computing Machinery et al., 2016) as the theoretical framework for interpretation. In this study, three approaches to implementing computer science were found: introducing computer science (a) as independent subjects, (b) within multiple subjects, and (c) as a part of transversal competencies or an independent computer science curriculum with a cross-curricular approach. We found common trends among the curricula of the 10 countries in how they implement the concepts and practices from the perspective of the scope and sequence. Most countries begin their curricula at the lower primary level, covering *algorithms*, *program development*, and developing computational artifacts along with the sub concepts under *impact of computing*. Then, countries tend to gradually introduce other concepts and practices as grade level goes up.

Since the introduction of computer science education at the primary level is relatively new, and the most of research conducted in this field is still limited, this study contributes to the worldwide efforts to introduce computer science education at the primary level. In addition, this study also supports both countries who have curricula and those who do not have curricula yet. Computer science curricula has characteristics of both longevity and changeability; therefore, even countries that have already implemented computer science education could be encouraged to review the trends of other countries' curricula in order to improve the curricula. For example, England revised their ICT curriculum to a computing curriculum in 2014 and reviewed the introduction status in 2017, as published in a report by the Royal Society (2017). In Australia, the next revised version will be published by the start of 2022 (Australian Government, Department of Education Skills and Employment, 2020) so that Australia can include recent trends in computer science and lessons learned from the current curriculum.

5.1 Limitations

There are four primary limitations of this study. Firstly, this study targeted national curricula for analysis but did not investigate the curriculum or syllabus that teachers use in their classrooms. There are three types of curriculum: (1) the intended curriculum, which is the body of written content that policymakers expect to be taught, (2) the curriculum that is taught, including the informal and formal lessons in classroom, and (3) the curriculum that students learn (Cuban, 1992). Porter & Smithson (2001) defined the taught curricula (also called enacted curriculum) as "the actual curricular content that students engage in the classroom" (p.2). Similarly, Falkner et al. (2019) argued that the taught curricula includes the contents that are delivered within the classroom and adopted pedagogical approaches. Thus, the taught curricula have deeply connected to common teaching that teachers adopted in each subject and included critical aspects that relate to subjects. In this regard, Cuban (1992) remarked that there is a gap between what is intended and what is taught. Therefore, research on the intended curricula, what knowledge and skills teachers deliver in the classroom, and the outcome of the students as a result of what they learn is crucial and lacking the study on the taught curricula could cause one to miss important aspects of the subject. This study focused on only the intended curriculum because of the limitation of this study and did not include the taught curriculum, which can include researching textbooks or syllabi, and the learned curriculum, as evident through students' assessments.

Secondly, this study set the criterion for selecting countries that had implemented country-wide computer science curricula in schools. Because of the methodological limitations of this study, we set the criterion and tried to gain the implications for primary computer science curriculum from maturing national curricula within 10 countries. However, this criterion could risk eliminating the computer science curricula that have been implemented in countries with different governmental structures such as Cantons and regional governments, although they are more populous than some of the nations that this study included. In the future, we would like to expand criterion to these countries and areas to study their computer science education curricula.

Thirdly, this study targeted not all subjects within the curricula, but instead focused only on the subjects that were related to integrated computer science; all curricula were published by government or equivalent institutions. If we investigated all subjects within the 10 countries' curricula, including both computer science and unrelated subjects, the results may have changed.

Finally, Google Translate was utilized to translate some of the curriculum texts from the original language to English. Since this study targeted diverse countries, we thought that this translation method was the best way to achieve both feasibility and reliability based on the research by Li et al. (2014) that investigated the reliability of using Google Translate by comparing Google Translate with human translation, finding that Google English translation showed a high correlation with both human English translation and an original Chinese text. However, the further development of translation tools such as Google Translate may further contribute to the accuracy of obtaining information and promoting comparative curriculum research in the future.

Although this study identified trends in K–12 computer science curricula by conducting a cross-curricular analysis, the detailed analysis integrating the perspective of the educational system and history could contribute to the development of high-quality computer science curricula in primary school, since curriculum reflects each country's policy, history, and culture, future research could include the analysis of educational systems in terms of computer science curriculum, focusing on the few nations that have developed high- quality curricula among the 10 countries studied here.

References

Association for Computing Machinery. (2014). Rebooting the Pathway to Success: Preparing Students for Computing Workforce Needs in the United States. *ACM Pathways Report, Pathways.Acm.Org*. http://pathways.acm.org

Association for Computing Machinery, Code.org, Computer Science Teachers Association, Cyber Innovation Center, & Math and Science Initiative. (2016). *K–12 Computer Science Framework*. http://www.k12cs.org

Australian Curriculum, Assessment and Reporting Authority. (2015). *Digital Technologies: Sequence of content F-10*. Retrieved from https://www.australiancurriculum.edu.au/f-10-curriculum/technologies/digital-technologies/pdf-documents/

Australian Government, Department of Education Skills and Employment. (2020, June 12). *Australian Curriculum*. Retrieved from https://www.education.gov.au/australian-curriculum-0

Australian Trade and Investment Commission. (n.d.). *Australian education system*. Elcom Technology. Retrieved September 5, 2020, from https://www.studyinaustralia.gov.au/english/australian-education/education-system

Baron, G.-L., Drot-Delange, B., Grandbastien, M., & Tort, F. (2014). Computer Science Education in French Secondary Schools: Historical and Didactical Perspectives. *Trans. Comput. Educ.*, *14*(2), 11:1–11:27. https://doi.org/10.1145/2602486

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kampylis, P., & Punie, Y. (2016). Developing Computational Thinking in Compulsory Education. Implications for policy and practice. *EUR - Scientific and Technical Research Reports*. https://doi.org/10.2791/792158

Bocconi, S., Chioccariello, A., & Earp, J. (2018). The Nordic approach to introducing computational thinking and programming in compulsory education. *Report prepared for the Nordic@BETT2018 Steering Group.* https://doi.org/10.17471/54007

Choi, J., An, S., & Lee, Y. (2015). Computing Education in Korea—Current Issues and Endeavors. *Trans. Comput. Educ.*, *15*(2), 8:1–8:22. https://doi.org/10.1145/2716311

Cohen, L., Manion, L., & Morrison, K. (2007). *Research Methods in Education, 6th Ed* (Vol. 1–6th Edition). Routledge.

Comer, D. E., Gries, D., Mulder, M. C., Tucker, A., Turner, A. J., & Young, P. R. (1989). Computing As a Discipline. *Commun. ACM*, *32*(1), 9–23. https://doi.org/10.1145/63238.63239

Computing at School. (2013). *Computing in the national curriculum A guide for primary teacher*. Retrieved from http://www.computingatschool.org.uk/data/uploads/CASPrimaryComputing.pdf

Computing at School Working Group. (2009).

Cuban, L. (1992). Curriculum Stability and Change. In P. W. Jackson (Ed.), *Handbook of Research on Curriculum: A Project of the American Educational Research Association* (pp.216-247).

Denning, P. J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM*, *60*(6), 33–39. https://doi.org/10.1145/2998438

Department for Education. (2013). *Computing programmes of study: Key stages 1 and 2 National curriculum in England*.

Directorate-General for Education, & Government of the Portuguese Republic. (2018). *Currículo Nacional—DL 55/2018* [National Curriculum—DL 55/2018]. Retrieved from https://www.dge.mec.pt/curriculo-nacional-dl-552018

ECDL Foundation. (2015). *Computing and Digital Literacy: Call for a holistic approach*. 9.

Ediger, M. (1995). Sequence and scope in the curriculum. *Education*, *116*(1), 159.

Education Development Center, & Ministry of Education. (2019, December 23). *Podstawa programowa kształcenia ogólnego dla liceum, technikum i branżowej szkoły II stopnia* [The core curriculum of general education for high schools, technical secondary schools and second-cycle industry schools]. Retrieved from https://www.ore.edu.pl/2018/03/podstawa-programowa-ksztalcenia-ogolnego-dla-liceum-technikum-i-branzowej-szkoly-ii-stopnia/

European Commission. (2017a, October 10). *Portugal Overview*. Eurydice - European Commission. Retrieved from https://eacea.ec.europa.eu/national-policies/eurydice/content/portugal_en

European Commission. (2017b, October 10). *Sweden Overview*. Eurydice - European Commission. Retrieved from https://eacea.ec.europa.eu/national-policies/eurydice/content/sweden_en

European Commission. (2019). *Poland-Organisation of the Education System and of its Structure*. Eurydice - European Commission. Retrieved from https://eacea.ec.europa.eu/national-policies/eurydice/content/organisation-education-system-and-its-structure-56_en

European Schoolnet. (2015). *Computing our future: Computer programming and coding—Priorities, school curricula and initiatives across Europe*.

Falkner, K., Sentance, S., Vivian, R., Barksdale, S., Busuttil, L., Cole, E., Liebe, C., Maiorana, F., McGill, M. M., & Quille, K. (2019). An International Comparison of K-12 Computer Science Education Intended and Enacted Curricula. *Proceedings of the 19th Koli Calling International Conference on Computing Education Research*, 1–10. https://doi.org/10.1145/3364510.3364517

Fayer, S., Lacey, A., & Watson, A. (2017). *STEM Occupations: Past, Present, And Future*. U.S. BUREAU OF LABOR STATISTICS.

French Academy of Sciences. (2013). *Teaching computer science in France Tomorrow can't wait*.

Gal-Ezer, J., & Stephenson, C. (2014). A Tale of Two Countries: Successes and Challenges in K-12 Computer Science Education in Israel and the United States. *ACM Transactions on Computing Education*, *14*(2), 8:1–8:18. https://doi.org/10.1145/2602483

Gander, W., Antoine, P., Gérard, B., Barbara, D. G., Jan, V., Andrew, M., Avi, M., & Chris, S. (2013). *Informatics Education: Europe cannot afford to miss the boat*.

Goodland, J. I., & Su, Z. (1992). Organization of the Curriculum. In P. W. Jackson (Ed.), *Handbook of Research on Curriculum: A Project of the American Educational Research Association* (pp.327-344).

GOV.UK. (n.d.). *The national curriculum*. Retrieved August 22, 2020, from https://www.gov.uk/national-curriculum

Gueudet, G., Bueno-Ravel, L., Modeste, S., & Trouche, L. (2017). Curriculum in France: A National Frame in Transition. In D. Thompson, M. A. Huntley, & & C. Suurtamm (Eds.), *International Perspectives on Mathematics Curriculum* (pp. 41–70). International Age Publishing. https://hal.archives-ouvertes.fr/hal-01599059

Heintz, F., Mannila, L., & Farnqvist, T. (2016). A review of models for introducing computational thinking, computer science and computing in K-12 education. *2016 IEEE Frontiers in Education Conference (FIE)*, 1–9. https://doi.org/10.1109/FIE.2016.7757410

Korean Ministry of Education. (2015). *고등학교 교육과정(I,II,III)* [High School Curriculum (I, II, III)].

Li, H., Graesser, A. C., & Cai, Z. (2014). *Comparison of Google Translation with Human Translation*. 6.

Li, Y., Schoenfeld, A. H., diSessa, A. A., Graesser, A. C., Benson, L. C., English, L. D., & Duschl, R. A. (2020). Computational Thinking Is More about Thinking than Computing. *Journal for STEM Education Research*, *3*(1), 1–18. https://doi.org/10.1007/s41979-020-00030-2

Madaus, G. F., & Kellaghan, T. (1992). Curriculum Evaluation and Assessment. In P. W. Jackson (Ed.), *Handbook of Research on Curriculum: A Project of the American Educational Research Association* (p. pp.119-154).

Maker, C. J. (1986). *Developing Scope and Sequence in Curriculum*. 8.

Ministry of Education. (2017). *Technology in the New Zealand Curriculum*.

Ministry of Education, Culture, Sports, Science and Technology. (2015). *諸外国におけるプログラミング教育に関する調査研究* [Research on Programming Education in International Countries].

Ministry of National Education and Youth. (2018, November 13). *Numérique, technologie, sciences informatiques [Digital, technology, computer science]*. Retrieved from https://eduscol.education.fr/maths/actualites/actualites/article/numerique-technologie-sciences-informatiques.html

National Board of Education. (n.d.). *Perusopetuksen opetussuunnitelmien perusteet* [Fundamentals of basic education curricula]. Opetushallitus. Retrieved January 26, 2020, from https://www.oph.fi/fi/koulutus-ja-tutkinnot/perusopetuksen-opetussuunnitelmien-perusteet

National Center on Education and the Economy. (n.d.). South Korea: Learning Systems. *NCEE*. Retrieved from http://ncee.org/what-we-do/center-on-international-education-benchmarking/top-performing-countries/south-korea-overview/south-korea-instructional-systems/

National Research Council. (2012). *A Framework for K-12 Science Education: Practices, Crosscutting Concepts, and Core Ideas*. https://doi.org/10.17226/13165

New Zealand Ministry of Education. (n.d.). *Technology in the NZC*. Retrieved January 29, 2020, from http://technology.tki.org.nz/Technology-in-the-NZC

POLISH REPUBLIC. (2017, February 24). *Rozporządzenie Ministra Edukacji Narodowej z dnia 14—Dziennik Ustaw* [Regulation of the Minister of National Education of 14—Journal of Laws]. http://dziennikustaw.gov.pl/du/2017/356

Rolandsson, L., & Skogh, I.-B. (2014). Programming in School: Look Back to Move Forward. *Trans. Comput. Educ.*, *14*(2), 12:1–12:25. https://doi.org/10.1145/2602487

Porter, A. C., & Smithson, J. L. (2001). Defining, Developing, and Using Curriculum Indicators. CPRE Research Report Series. https://eric.ed.gov/?id=ED477657

Schmidt, W. H., McKnight, C. C., Valverde, G., Houang, R. T., & Wiley, D. E. (1997). *Many Visions, Many Aims: A Cross-National Investigation of Curricular Intentions in School Mathematics*. Springer Science & Business Media.

Schmidt, William H., Wang, H. C., & McKnight, C. C. (2005). Curriculum coherence: An examination of US mathematics and science content standards from an international perspective. *Journal of Curriculum Studies*, *37*(5), 525–559. https://doi.org/10.1080/0022027042000294682

Skolverket. (2017). *Få syn på digitaliseringen på gymnasial nivå* [Get an idea of digitalisation at upper secondary level]. Retrieved from https://www.skolverket.se/publikationsserier/kommentarmaterial/2017/fa-syn-pa-digitaliseringen-pa-gymnasial-niva?id=3784

So, H.-J., Jong, M. S.-Y., & Liu, C.-C. (2020). Computational Thinking Education in the Asian Pacific Region. *The Asia-Pacific Education Researcher*, *29*(1), 1–8. https://doi.org/10.1007/s40299-019-00494-w

Tenenberg, J., & McCartney, R. (2014). Editorial: Computing Education in (K-12) Schools from a Cross-National Perspective. *ACM Trans. Comput. Educ.*, *14*(2), 6:1–6:3. https://doi.org/10.1145/2602481

The Curriculum Development Council. (2017a). *General Studies Curriculum Guide for Primary Schools (Primary 1 – Primary 6)*.

The Curriculum Development Council. (2017b). *Mathematics Education Key Learning Area Curriculum Guide (Primary 1 – Secondary 6)*.

The Curriculum Development Council. (2017c). *Technology Education Key Learning Area Curriculum Guide (Primary 1 – Secondary 6)*.

The Institute for Democracy. (2018). *소프트웨어 교육 현황과 개선 방향* [Software education status and improvement direction].

The Royal Society. (2012). *Shut down or restart? The way forward for computing in UK schools*. https://royalsociety.org/topics-policy/projects/computing-in-schools/report/

The Royal Society. (2017). *After the reboot: Computing education in UK schools*. https://royalsociety.org/topics-policy/projects/computing-education/

Wing, J. (2010). Computational Thinking: What and Why? *Unpublished Manuscript in Progress.* Referenced in http://www.cs.cmu.edu/~CompThink/papers/TheLinkWing.pdf.

Appendix A

Primary Sources of the National Curricula in This Study

Country	Curricula
Australia	Digital Technologies: Sequence of content F-10 (Australian Curriculum, Assessment and Reporting Authority, 2015)
England (United Kingdom)	Computing programmes of study: key stages 1 and 2 National curriculum in England (Department for Education, 2013) Computing programmes of study: key stages 3 and 4 National curriculum in England (Department for Education, 2013)
Finland	Perusopetuksen Opetussuunnitelman Perusteet 2014 [Basic Education Basics of the Curriculum 2014] (National Board of Education, 2014) Lukion Opetussuunnitelman Perusteet 2015 [Upper School Basics of the Curriculum 2015] (National Board of Education, 2015)
France	Programme du cycle 2 En vigueur à compter de la rentrée de l'année scolaire 2018-2019 [Cycle 2 Program Effective as of the beginning of the 2018-2019 school year] (Ministry of National Education and Youth, 2018) Programme du cycle 3 En vigueur à compter de la rentrée de l'année scolaire 2018-2019 [Cycle 3 Program Effective as of the beginning of the 2018-2019 school year] (Ministry of National Education and Youth, 2018) Programme du cycle 4 En vigueur à compter de la rentrée de l'année scolaire 2018-2019 [Cycle 4 Program Effective as of the beginning of the 2018-2019 school year] (Ministry of National Education and Youth, 2018) Sciences numériques et technologie Classe de seconde, enseignement commun [Digital Sciences and Technology Second Class, Common Teaching] (The Higher Program Council (CSP), 2018) Mathématiques Classe de seconde, enseignement commun [Mathematics Second Class, Common Teaching] (The Higher Program Council (CSP), 2018) Numérique et sciences informatiques Classe de première, enseignement de spécialité, voie générale [Digital and Computer Sciences First Class, Specialty Education, General Track] (The Higher Program Council (CSP), 2018) Mathématiques Classe de première, enseignement de spécialité [Mathematics First Class, Specialty Education] (The Higher Program Council (CSP), 2018)
Hong Kong	General Studies Curriculum Guide for Primary Schools (Primary 1 – Primary 6) (The Curriculum Development Council, 2017)

	Technology Education Key Learning Area Curriculum Guide (Primary 1 – Secondary 6) (The Curriculum Development Council, 2017)
Korea	초등학교 교육과정 교육부 고시 제 2015-74 호 [별책 2] [Elementary School Curriculum and Education Notice No. 2015-74] (Ministry of Education, 2015) 중학교 교육과정 교육부 고시 제 2015-74 호 [별책 3] [Middle School Curriculum and Education Notice No. 2015-74] (Ministry of Education, 2015) 고등학교 교육과정(I,II,III) [High School Curriculum (I, II, III)]. (Ministry of Education, 2015)
New Zealand	Technology in the New Zealand Curriculum (Ministry of Education, 2017)
Poland	Podstawa programowa kształcenia ogólnego z komentarzem. Szkoła podstawowa Informatyka [General Education Core Curriculum with Commentary. Elementary School Computer Science] (Education Development Center, and Ministry of Education, 2017) Podstawa programowa kształcenia ogólnego z komentarzem. Szkoła ponadpodstawowa: liceum ogólnokształcące, technikum oraz branżowa szkoła I i II stopnia Informatyka [General Education Core Curriculum with Commentary. Secondary School: General High School, Technical High School and Industry First- and Second-Degree Computer Science] (Ministry of Education, 2019)
Portugal	1.º CICLO DO ENSINO BÁSICO ORIENTAÇÕES CURRICULARES PARA AS TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO [1st Basic Education Cycle Curriculum Guidelines for Information and Communication Technologies] (Directorate-General for Education, and Government of the Portuguese Republic, 2018) 5.º ANO | 2.º CICLO DO ENSINO BÁSICO TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO [5th Year | 2nd Basic Education Cycle Information and Communication Technologies] (Directorate-General for Education, and Government of the Portuguese Republic, 2018) 6.º ANO | 2.º CICLO DO ENSINO BÁSICO TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO [6th Year | 2nd Basic Education Cycle Information and Communication Technologies] (Directorate-General for Education, and Government of the Portuguese Republic, 2018) 7.º ANO | 3.º CICLO DO ENSINO BÁSICO TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO [7th Year | 3rd Basic Education Cycle Information and Communication Technologies] (Directorate-General for Education, and Government of the Portuguese Republic, 2018) 8.º ANO | 3.º CICLO DO ENSINO BÁSICO TECNOLOGIAS DA INFORMAÇÃO E

	COMUNICAÇÃO [8th Year | 3rd Basic Education Cycle Information and Communication Technologies] (Directorate-General for Education, and Government of the Portuguese Republic, 2018) 9.º ANO | 3.º CICLO DO ENSINO BÁSICO TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO [9th Year | 3rd Basic Education Cycle Information and Communication Technologies] (Directorate-General for Education, and Government of the Portuguese Republic, 2019) 12.º ANO | ENSINO SECUNDÁRIO APLICAÇÕES INFORMÁTICAS B [12th Year | Secondary Education Computer Applications B] (Directorate-General for Education, and Government of the Portuguese Republic, 2018)
Sweden	Läroplan för grundskolan, förskoleklassen och fritidshemmet [Curriculum for elementary school, Preschool Class and Kindergarten] (National Agency for Education (Skolverket), 2019) (English version exists.) Swedish, Swedish as a second language, Social studies, Technology, Information and communication, Computers and ICT (National Agency for Education (Skolverket), n.d., Subject plans in upper secondary school in English)

Appendix B

Concepts of K–12 Computer Science Framework

Core Concepts	Sub Concepts
Computing Systems	Devices
	Hardware and Software
	Troubleshooting
Networks and the Internet	Network Communication and Organization
	Cybersecurity
Data and Analysis	Collection
	Storage
	Visualization and Transformation
	Inference and Models
Algorithms and Programming	Algorithms
	Variables
	Control
	Modularity
	Program Development
Impact of Computing	Culture
	Social Interactions
	Safety, Law, and Ethics

Appendix C

Practices of K–12 Computer Science Framework

Practices	By the end of Grade 12, students should be able to
1. Fostering an Inclusive Computing Culture	1. Include the unique perspectives of others and reflect on one's own perspectives when designing and developing computational products.
	2. Address the needs of diverse end users during the design process to produce artifacts with broad accessibility and usability.
	3. Employ self- and peer-advocacy to address bias in interactions, product design, and development methods.
2. Collaborating	1. Cultivate working relationships with individuals possessing diverse perspectives,

Around Computing	skills, and personalities.
	2. Create team norms, expectations, and equitable workloads to increase efficiency and effectiveness.
	3. Solicit and incorporate feedback from, and provide constructive feedback to, team members and other stakeholders.
	4. Evaluate and select technological tools that can be used to collaborate on a project.
3. Recognizing and Defining Computational Problems	1. Identify complex, interdisciplinary, real-world problems that can be solved computationally.
	2. Decompose complex real-world problems into manageable subproblems that could integrate existing solutions or procedures.
	3. Evaluate whether it is appropriate and feasible to solve a problem computationally.
4. Developing and Using Abstractions	1. Extract common features from a set of interrelated processes or complex phenomena.
	2. Evaluate existing technological functionalities and incorporate them into new designs.
	3. Create modules and develop points of interaction that can apply to multiple situations and reduce complexity.
	4. Model phenomena and processes and simulate systems to understand and evaluate potential outcomes.
5. Creating Computational Artifacts	1. Plan the development of a computational artifact using an iterative process that includes reflection on and modification of the plan, taking into account key features, time and resource constraints, and user expectations.
	2. Create a computational artifact for practical intent, personal expression, or to address a societal issue.
	3. Modify an existing artifact to improve or customize it.
6. Testing and Refining Computational Artifacts	1. Systematically test computational artifacts by considering all scenarios and using test cases.
	2. Identify and fix errors using a systematic process.
	3. Evaluate and refine a computational artifact multiple times to enhance its performance, reliability, usability, and accessibility
7. Communicating About Computing	1. Select, organize, and interpret large data sets from multiple sources to support a claim.
	2. Describe, justify, and document computational processes and solutions using appropriate terminology consistent with the intended audience and purpose.

	3. Articulate ideas responsibly by observing intellectual property rights and giving appropriate attribution.

Appendix D

Analysis Contents from the National Curricula in This Study

Country	Contents
Australia	Sequence of content F-10 Strand: Knowledge and understanding (Digital Technologies, F–10)
England (United Kingdom)	Subject content (Computing, Years 1–11)
Finland	Matematiikan tavoitteisiin liittyvät keskeiset sisältöalueet [Key content areas related to mathematics objectives] (Mathematics, Grades 1–9) Käsityön tavoitteisiin liittyvät keskeiset sisältöalueet [Key content areas related to craft objectives] (Crafts, Grades 3–9) Laaja-alainen osaaminen [Extensive expertise] (Transversal competencies (ICT competences), Grades 1–9)
France	"Nombres et calculs [Numbers and calculations]" and "Espace et géométrie [Space and geometry]" (Mathematics, Grades 1–6), Connaissances, Compétences associées [Knowledge, Associated Skills] (Mathematics, Grades 7–10), Histoire des mathématiques [History of Mathematics] and Capacités associées [Associated Capabilities] in "Algorithmique et programmation [Algorithms and programming]", (Mathematics, Grades 11) Connaissances et compétences associées [Associated knowledge and skills] (Science and Technology, Grades 4–6) Connaissances et compétences associées [Associated knowledge and skills] (Technology, Grades 7–9) Contenus, Capacités attendues [Content, Expected Capacities] (Digital Science and Technology, Grades 10) Contenus, Capacités attendues [Content, Expected Capacities] (Digital and Computer Science, Grade 11)
Hong Kong	Knowledge and Understanding, Skills, Values and Attitudes in "Strand 3 (Science and Technology in Everyday Life)" and "Strand 6 (Global Understanding and the Information Era)", (General Study, Primary 1–6)

	Knowledge contexts (Information and Communication Technology (ICT)), Modules (K1 Computer Systems, K2 Programming Concepts, K16 Information Processing and Presentation, and E1 Computer Networks), and Content (Technology, Secondary 1–3) Learning Outcomes and Remarks (Information and Communication Technology, Secondary 4–6)
Korea	성취기준 [Achievement standards] (Practical Arts (Technology/ Home Economics, Primary 5, 6) 성취기준 [Achievement standards] (Informatics, Lower and Upper Secondary 1–3)
New Zealand	Progress outcome in "Computational thinking for digital technologies" and "Designing and developing digital outcomes" (Technology, Years 1–13)
Poland	Edukacja informatyczna [Information technology education] (Informatics, Grades 1–3) Treści nauczania – wymagania szczegółowe [Teaching content - specific requirements] (Informatics, Grades 4–11)
Portugal	CONHECIMENTOS, CAPACIDADES E ATITUDES [Knolwedge, Capacity, and Attitude] (ICT, Grades 1–9, Computer Applications B, Grade 12)
Sweden	Centralt innehåll [Core content] (Mathematics, Grades 1–9) Centralt innehåll [Core content] (Technology, Grades 1–9) Centralt innehåll [Core content] (Social Studies, Grades 4–9) Core content (Social Studies, Grades 10–12) Central content (Swedish / Swedish as a second language, Grades 10–12) Core content (Computers and ICT, Grades 10–12) Core content (Information and communication, Grades 10–12) Core content (Technology, Grades 10–12)

Investigating Pre-School Children's Perspectives of Robots through Their Robot Drawings[1]

Ela Sümeyye SEÇİM[1]

Mine Canan DURMUŞOĞLU[2]

Mustafa ÇİFTÇİOĞLU[3]

[1]Amasya University

[2]Hacettepe University

[3]Pamukkale University

DOI:10.21585/ijcses.v4i4.112

Abstract

This study investigated preschool children's opinions on educational robots using their robot drawings. The study group consisted of 64 five- and six-years old children in an independent kindergarden affiliated to the Ministry of National Education (MoNE) in Ankara province, Turkey and participated in the Preschool Robotics Coding Workshop within the scope of the project titled "TUBITAK 4007 Science in the Footsteps of Cezeri" in the 2018-2019 academic year. For the purpose of this study phenomenology model as a qualitative research approach was adopted and data was collected through visual materials and semi-structured interviews to determine children's opinions on robots. The data collection process was carried out in an eight-weeks period starting in March 2019 as one week for the acquaintance phase, six weeks for practice, and the last week for a two-day workshop with educational robots. In line with the findings of this research, the mechanical features of the robots were examined, it was determined that there was an increase in the battery drawings in children's last drawings compared to their first drawings. It was also determined that the children responded to "Who builds the robots?" interview question as factories, scientists, machines, and repairers in the first interview while more than 90% of the children responded as scientists and engineers in the second interview.

Keywords: Child drawings, educational robot, preschool children, robot drawings, coding

[1] This article was produced from TUBITAK 4007 Science in the Footsteps of Cezeri Project.

1. Introduction

Painting art is a type of creative activity for expressing thoughts, goals, phenomena, and events using imagination and transferring them to others (Lowenfeld, 1971). Children can also transfer many thoughts that they cannot verbally express through drawing pictures. These drawings, which appear as unconscious scribbles in the early periods, gain meaning over time. These meaningful drawings provide clues to the child's exploration of the world (İskenderoğlu, 2006; Metin & Aral, 2012). When the related literature is examined, it is seen that the development stage of drawing in childhood is divided into five stages. These stages are: Scribble Stage (2 to 4 years), Preschematic Stage (4 to 7 years), Schematic stage (7 to 9 years), Dawning Realism Stage (9 to 11 years), and the Pseudorealistic Stage (12 to 14 years) (Lowenfeld, 1971).

In the Preschematic Stage covering the 4 to 7 years and including the preschool education period after the scribble stage, it is emphasized that it is important to examine the pictures of children as they are important in terms of their drawing development. When the characteristics of this stage are examined, it is understood that the children start to use symbols in this stage. The child can transfer the reflection of the objects he/she encounters in his/her daily life to his/her drawings by using different materials (Gürtuna, 2004; Malchiodi, 2005; Metin & Aral, 2012).

According to the previous study, the images seen in children's drawings are important data resources about them, but they are not sufficient alone (Ersoy & Türkkan, 2009), and therefore the data obtained from the literature emphasize the necessity for using interviews to support children's drawings. Helping children who have difficulties in expressing themselves talk about their drawings is effective in triggering their memory and recalling more memories and information (Sayıl, 2004). Based on this information, it can be inferred that children can express their feelings and experiences more clearly in their drawings.

2. The Study

2.1. Aim and Significance of the Study

This study aims to investigate the robot drawings of five-six years old children and their opinions on educational robots in the world surrounded by technology. While there were studies which focused on the use of education robots in education, there were no studies investigated the opinions of children on educational robots. According to the previous research, people live in a world surrounded by technology (Bers, 2008). From the pen we use to the phone, to the computer and camera, the objects around us change and evolve with the technology. For example, the tap automatically opens when we want to wash our hands, elevator door does not close even if there is a small object in the doorway, our phones know how to send electronic mails, and what time it should wake us up (Bers & Horn, 2010). The new generation, which was born in the 21st century and grew with the power to control and develop all these objects, is called "the digital natives" because of their proximity and predisposition to the digital world (Prensky, 2001).

One of the tools used as educational materials in the classes are educational robots. These robots used in the educational process are preferred as they give immediate feedback about the targeted concept and are found interesting by the children. If the interest levels of the students are high, their motivation and desire to learn also increase (Resnick, 2003). However, there are some difficulties in using educational robots in the training process.

Teachers' not having sufficient knowledge and experience in this regard, and teacher training programs' not training the teacher candidates with a vision of following technological developments are some of these difficulties (Bers et all., 2002; Kasalak, 2017). According to the research, it is necessary to increase the number of studies to be carried out on this subject and to provide in-service training opportunities for teachers to include educational robots in education programs (Ioannidou, et all., 2011; Johnson, 2003).

In addition to the use of educational robots in education as a part of 21st-century skills, it is also very important to provide coding skills for children. Despite all these, it was revealed as a result of the research that many teachers working in the field had little experience on this subject and some teachers had almost no experience. It was also mentioned that the programs which train preschool teachers do not train teacher candidates with a vision of following technological developments. It is considered that this situation causes preschool teachers to choose the role of a teacher who consumes technology rather than choosing the role of a teacher who designs a technologically enriched program (Bers et all., 2002). In addition to this, it was also mentioned in the related literature that the coding process is an abstract concept, it could be objectified for the children by dramatization activities, and the lack of dramatization activities constituted a problem situation (Odacı & Uzun, 2017). In light of the findings obtained from this study, the studies to be carried out on coding training in the preschool period should objectify the coding process through dramatization and minimize the disadvantage of children being illiterate by using visual materials. When teachers and students recognize the educational robots, they can use in robotics coding and perform activities in the classes with these materials, the interest and motivation of students will increase and teachers' hesitation about using materials will disappear.

This study aims at determining the opinions of children born in a technological world on the educational robots used in the classes. This study is significant in terms of serving as a model for similar studies, increasing the usage competencies and motivations of preschool teachers to use educational robots to enrich their educational statuses, considering the opinions of children in robot designs by the ones working in the field of educational robot design, and contributing to the relevant literature.

1. How are the perceptions of children about educational robots?
2. How are the educational robot experiences of children?
3. What is the difference between the first and last drawings of children?
4. Which activity was most liked by the children?

This research aims to find answers to these questions. Determining children's perceptions of robots will be useful for designers who design robots. It will also give teachers an idea about choosing an educational robot that they plan to use in the classroom. Determining the educational robot experiences of children in the learning process will help to prevent problems that may occur in the teaching process that will be planned in the future. It will enable teachers to have an idea about children's views and to take precautions against problems that may occur in the learning and experiencing process of educational robots. Examining the differences between the first and last drawings of children is an opportunity to examine the impact of the educational robot experience process on them in a detail way. It is thought that investigating the most liked activity of children is necessary in terms of forming a basis for teachers' activity planning and choices. It will show teachers how educational robot experience is from children's eyes. This research will enable teachers to gain insight into children's educational robot experiences, perspectives, and drawings. In addition to this, it will contribute to the literature on educational robot design, educational robot activities and children's drawings.

2.2. Relevant Scholarship

Examining the drawings of children has been the choice of many scientists with expertise in different fields. It is understood from the literature that the drawings of children have been adopted as research subjects in different subject fields. In this regard, some of the studies encountered during the literature review are as follows: Studies for determining the perceptions of children towards the concept related to the world and daily life (Klein, 1982; Ehrlen, 2009; Chang, 2012); studies for determining the relationships between children's emotional development and mother drawings (Cox & Moore, 1994; Çakmak & Darıca, 2012); studies for determining the emotional states of children based on the sizes, colors, and facial expressions of the figures drawn by the children (Beck& Feldman, 1989; Burkitt, Barret & Davis, 2009); studies for determining the scientist image in children's mind (Buldu, 2006); studies for determining the scientist perceptions of children (Güler & Akman, 2006); studies for determining children's perceptions about environmental problems (Barraza, 1999; Sadık, Çakan & Artut, 2011); studies for revealing the mathematics teacher images of primary school students (Picker & Berry, 2000); studies for determining how the concept of health is perceived by children (Rijey & Van Rooy, 2007); studies examining mythological drawings (Pehlivan, 2008); studies for determining the first sacred ceremony perceptions of children (Stokrocki &Kırışoğlu, 1996), and studies on family life (Türkkan, 2004) are some of the studies aimed at explaining a particular event, phenomenon, or object based on the drawings of children. When the national literature was reviewed, it was seen that there was no comprehensive study on children's robot drawings. When international literature was reviewed, it was seen that a study was carried out to examine children's opinions and drawings. In that study, children were asked to examine the robot drawings (Woods & Joerg, 2004).

2.3. Method

This section includes information about the study group, data collection tools, data collection process, and data analysis.

This research has been designed according to the phenomenology model, one of the qualitative research approaches. Phenomenology model is used to explain the common meaning of phenomenon or concept one or more people's experiences (Creswell, 2013). Phenomenology focuses on how people perceive phenomena, how they describe them, how they feel about it, how they judge and remember them, how they make sense, and how they talk to others about it (Patton, 2014). Phenomenological studies focus entirely on individual perceptions and try to investigate the experience from the perspective of the "inside" (Tanyaş, 2014). In this study, investigating pre-school children's perspectives of robots through their robot drawings were determined as a phenomenon. The visual data obtained from children's drawing was analyzed through document review while the interview forms were analyzed through descriptive analysis.

2.4. Sample

The criteria sampling method was used in this study. In this regard, the determined criteria were as follows: continuing four different age groups in an independent kindergarten affiliated to the MoNE in Etimesgut district of Ankara province, Turkey, being in the range of 48 to 72 months old and attending the Preschool Robotics Coding Workshop within the scope of the project titled "TUBITAK 4007 Science in the Footsteps of Cezeri". The study group of this study included 43 male children and 21 female children (total of 64). The frequency distributions of the ages and genders of children in the study group according to the classes were presented in Table 1. In this regard, the classes in the independent kindergarten were coded as "Class A, B, C, and D" in Table 1.

Table 1. Sample

Child Properties		**Class A**	**Class B**	**Class C**	**Class D**	**Total**
Age	48 to 60 months old	9	9	9	7	**34**
	60 to 72 months old	8	5	10	7	**30**
Gender	Female	7	4	7	3	**21**
	Male	10	10	12	11	**43**

2.5. Data Collection Tools

Visual materials and semi-structured interview forms were used as data collection tools to determine children's opinions on educational robots. Visual materials are one of the qualitative data collection tools. They offer participants the opportunity to directly share their facts (Creswell, 2013). The visual materials in the research process consist of robot drawings drawn by the children participating in this study. In this study carried out within the scope of the "TUBITAK 4007 Science in the Footsteps of Cezeri" project, data were collected through children's drawings and semi-structured interview forms developed by the researchers. Information about how the data collection tools used in this study are created and how they are used in the research process are provided below.

2.5.1. Robot Drawing of Children

When the drawings of children are analyzed well, making drawings help to reveal the cognitive structures clearly even if the concepts investigated are complex and it is effective in revealing the schemes existing in the minds of children and their relationships with other schemes (Schafer, 2012). Therefore, the data were collected from the drawings made by children, and the expressions of children were noted behind these drawings by interviewing with each child about their drawings. In many studies carried out in the literature on drawing analysis, the uncertainty of the drawings of the preschool children revealed the need for verbal explanations when analyzing their drawings. Therefore, the opinions of children on the subject were also collected along with their drawings as data. Data regarding children's drawings were collected in two stages. In the first meeting after the acquaintance phase, the children in the four classes of the independent kindergarten were requested to make a robot drawing. After the educational practices were performed with the robots one hour a week, they were finally requested to make another robot drawing. The symbols that children included in their drawings and the messages and stories they wanted to tell by using these symbols were noted behind the drawings.

2.5.2. Semi-structured interviews

After the semi-structured interview form was developed by the researchers, it was reorganized and finalized by taking the opinions of three experts. Considering these three experts, one expert was from the child development and education field, one expert was from the preschool education field, and one expert was from the assessment and evaluation field. There were a total of nine questions in the semi-structured interview form. In the first meeting with the children during the interviews within the scope of this study, the following questions were asked: "What is a robot?", "Have you ever seen a robot?", "What do robots do?", "How do robots work?", "Who makes the robots?", "If you were a robot, what kind of robot would you be?", "Why would you want to be that robot?", "What do you need to make a robot that does what you want?", and "What is robotics coding activity?". At the end of six-week educational practices, a final interview was made using the same interview questions. The answers provided by the children regarding the interview questions were recorded on the child interview form. The first interview form consisted of nine questions while the second interview form consisted of 11 questions. The additional questions in the second interview form were as follows: "Which activity did you like the most?" and "Why?".

2.6. Data Collection Process

The legal permissions to conduct the study in an independent kindergarten were obtained from the Ministry of National Education, Turkey. In addition to this, the institutional permissions received from Hacettepe University and TUBITAK within the scope of "TUBITAK 4007 Science in the Footsteps of Cezeri" project were completed, and consent forms were obtained from both children and their parents since the participants of the study were under the age of 18. The data were collected in March and April months of the 2018-2019 academic year. The data collection process was carried out in an eight-week period starting in March 2018 as one week for the acquaintance phase with children and teachers, six weeks for practice, and the last week for a two-day workshop with educational robots. The data collection process was completed in four stages (Figure 1).

The activities were planned by the researchers. Before the activities were planned, a total of 4 field experts were interviewed, 1 expert in preschool program, 2 experts in robotics coding, 1 expert in child development. After these interviews, the researchers decided on the activities they would implement and the educational robots they would use. In addition to this, it was thought that it would be beneficial to use different educational robots during the education process.

In the first stage, before performing activities with the children in the first week, they were requested to make a robot drawing and their short stories were noted behind their drawings. Then, the first interview was held to determine their opinions on robots. Next, various activities were performed with educational robots in the classes for six weeks. The activities were performed by using educational robots that do not require a computer or any installation as much as possible. During these activities, mobile coding mats (carpets) consisting of 4 different robots and obstacles with functions from simple to the complex were used. When performing these activities, the children guessed how many steps would be needed for the robots to reach from the starting point to the endpoint without hitting the obstacles and they affixed the pre-prepared arrow signs on the path the robot would go on the coding ground. Then, they pressed the start button by using the robot's remote-control unit and moved the robot from the starting point to the end point by moving forward, backward, right, and left on the coding ground. After performing educational activities with children for seven weeks, the science festival was completed by performing preschool robotics coding activities with the children within the scope of "TUBITAK 4007 Science in the Footsteps of Cezeri" project in the last week. In the second stage, the children were asked to make a robot drawing for the second time and then, the second interviews about robots were held after completing the science festival titled "TUBITAK 4007 Science in the Footsteps of Cezeri".

Figure 1. The Steps Followed During the Project Process

First step: Gathering the first robot drawing of children and interviewing with children about robots and activities
Second Step: Performing various activities with educational robots for six weeks
Third Step: Children perform their activities in festival area
Last Step: Gathering the last robot drawing and interviewing with children

2.7. Studies Performed within the Scope of "TUBITAK 4007 Science in the Footsteps of Cezeri" Project

After making acquaintance with the teachers and children in the first week, the teacher training process was started. Within the scope of workshop works, information forms to be sent to the parents and teachers and voluntary participation consent forms were sent to the parents who wanted their children to participate in the study, and forms were retrieved. Then, parent volunteer participation approval/permission forms regarding the volunteer participation of their children in robotics coding activities and workshops were sent to parents. Before continuing with the use of educational robots for the first three weeks, games with position in space theme were played to reinforce the concepts of forward, backward, right, and left with the children. In this regard, some of the games played were as follows: A game in which a child directs an adult, who has a robot role, by giving commands such as turn right, two steps forward, turn left, two steps backward, etc., a game in which children with the role of an adult direct other children as if they were robots, and a game in which a child, who is "It", directs the other child to draw the pattern on the canvas by telling the instructions on the worksheet without showing it to the other child. In the following weeks, practices regarding the activities to be performed in the workshop and the use of four different educational robots (BeeBot, Evaluation, DocRobot, Bootly) were organized for the children and teachers, and the activities were performed. With the activities performed, it was aimed for children to acquire necessary skills such as matching, ranking, position in space, positional language, direction language, problem-solving, and being able to give directions before the robotics coding workshop to be carried out within the scope of "TUBITAK 4007 Science in the Footsteps of Cezeri" project. In this regard, it was aimed to objectify the coding process through games by using visual materials, educational robots, and different teaching methods and techniques before coding works. In addition to this, it was also aimed for children to internalize problem-solving, critical thinking, and algorithmic thinking. During the activities, particular importance was given for children to enjoy the activities. Thus, it was aimed that children would develop a positive attitude towards technology and science. In the six-week period in which the activities were performed, paper activities for improving the coding skills that children could do with their parents at home were sent to the parents as a family participation study. As in the ideal model that should be in preschool education quality standards, these activities were performed in cooperation with children, families, educators, and school administrators. After performing educational activities with children for seven weeks, preschool robotics coding activities were performed in the last week of project.

2.8. Data Analysis

The visual data obtained from the robot drawings of children were analyzed through the document review technique from qualitative research data analysis techniques and the data obtained from the interview forms were analyzed through a descriptive analysis technique. The document review is defined as the examination of all data at the macro and micro levels such as any document, image, or sound recording (Yıldırım & Şimşek, 2016). Descriptive analysis is a type of qualitative data analysis that includes summarizing and interpreting the data obtained through various data collection techniques according to the predetermined themes (Özdemir, 2010). In this regard, some themes were determined by considering the topics in the interview form. The questions in the semi-structured interview form were considered as themes in the descriptive analysis. The visual data obtained from the drawings were analyzed through the document review technique. The themes obtained from the document review technique were presented in Appendix 1. The drawings not included in the themes were added as themes later. To increase the reliability of the study and to ensure inter-coder consistency among the researchers, 10 independently selected drawings were coded separately by the researchers. The correlation coefficient was examined by considering the coding of both researchers. The correlation coefficient in the analyzed 10 drawings was found to be .98 ($p<0.001$) and it was determined that the consistency between the researchers was high. Achieving internal validity in qualitative research is ensured by eliminating the subjective perception of the researcher and examining the research subject as objectively as possible (Yıldırım & Şimşek, 2016).

3. Results

3.1. Findings and Interpretations

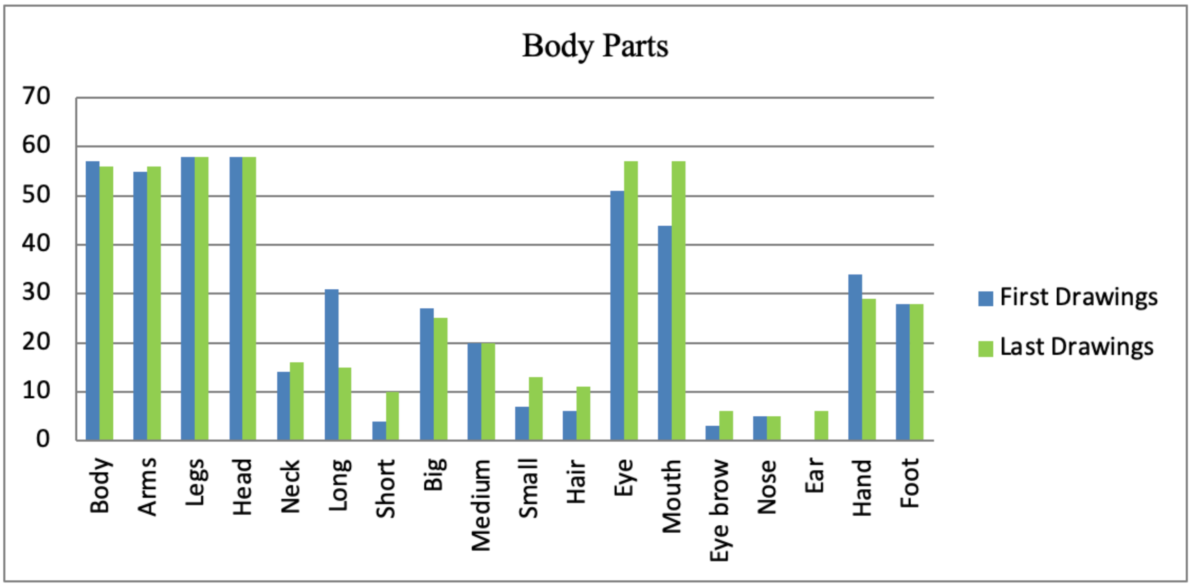
This section includes findings obtained from the children's first and last robot drawings and findings obtained from the first and last interviews. The findings considering the children's first and last drawings included the distribution graphics of the parts of the robot body, the line features used in the drawing, the colors used, the robot appearance feature, the robot emotion drawings, mechanical features of the robot, the shapes they used in the drawings, the availability of features for the things the robot does, and the way robots work.

In this part, it was stated which subtitle was examined. Then, the table of the examined title is included. After the table explanation, striking pictures of the children about the title were placed. For example, the first subhead is body parts. The situation of drawing the body parts in the first and last drawings of the children was shown with a table. After that the table explained. Then, sample pictures were added.

3.1.1. Findings Related to Children's Drawings: Body Parts

In the graphic in Table 2, it is seen that there is no remarkable difference between the children's first drawings and last drawings considering the body parts such as the body, arms, legs, head, neck, and feet. It can be inferred from this graphic that children generally attributed human features to the robots. When the sensor drawings between the first and last drawings were examined, it was seen that the number of eye and mouth drawings increased. Road maps used for children to find the paths of the robots they experienced with both the researchers and their teachers for six weeks were thought to cause children to think that robots had vision sensors. The fact that the robots used in the research process had speech characteristics can be associated with the increase in the mouth drawings in children's final drawings. In addition to this, it was seen that the first drawings did not include ears or similar organs related to hearing sensors while some drawings included hearing sensors. The fact that some robots used in this study had voice detection and voice command features can be associated with this finding. When the dimensions of the robot drawings were examined, it was seen that large and medium dimensions were the majority and small dimensions were the minority in the first drawings. Considering the last drawings, it was seen that the number of robots drawn with small dimensions decreased. In other words, it was observed that the robot dimension drawings enlarged with the educational process received.

Table 2. The distribution graphics of robot body parts in children's first and last drawings



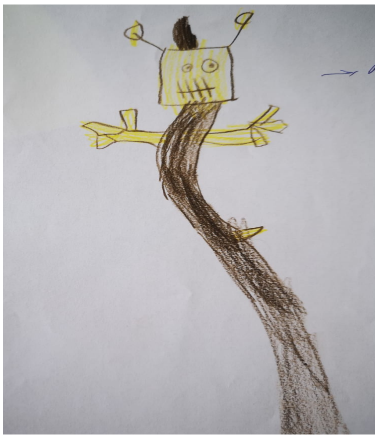


Figure 2. Drawing realized by a 5-year-old boy

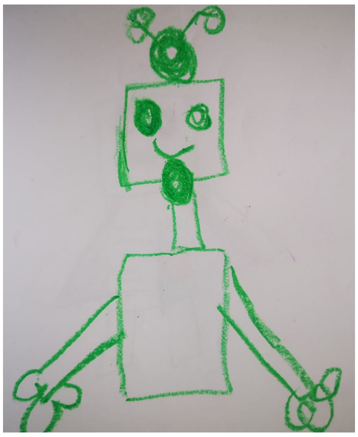


Figure 3. Drawing realized by a 4-year-old boy

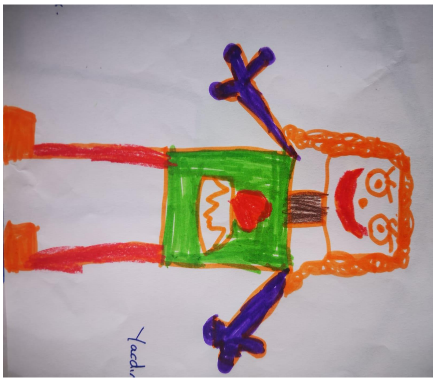


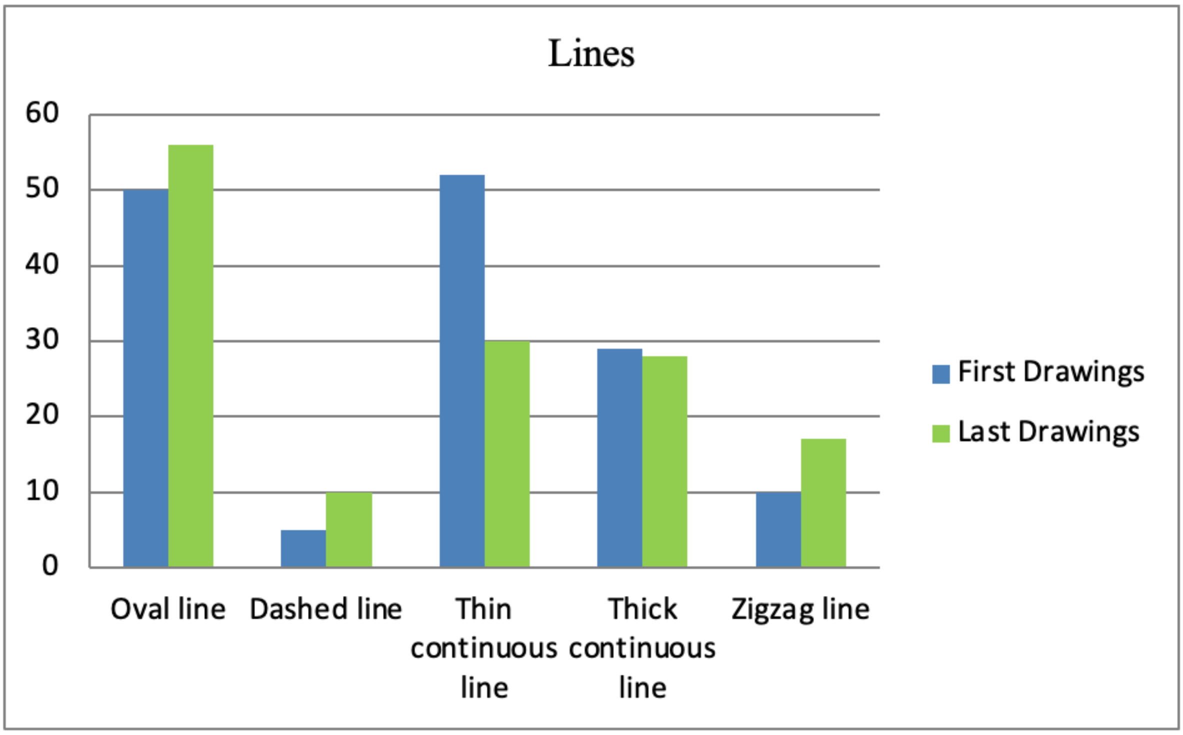
Figure 4. Drawing realized by a 5-year-old girl

*Figure 2-3-4 are placed to show how children expressed the body parts of robots in their drawings.

3.1.2. Findings Related to Children's Drawings: Lines

As can be seen in Table 3, the lines used in children's robot drawings consist mostly of oval lines, thin continuous lines, and thick continuous lines respectively. It can be inferred from the table that oval lines were used to draw the head of the robot rather than other parts of the body, and thin and thick lines were mostly used in the body parts. It was determined that the use of zigzag lines increased in the last drawings.

Table 3. The distribution graphics of the **lines** used in children's first and last drawings



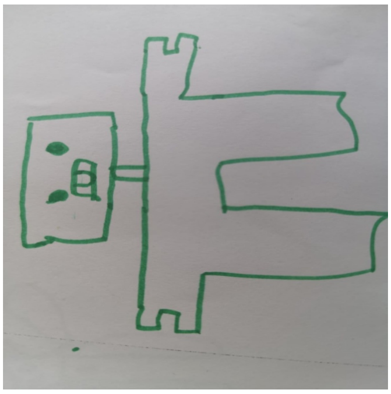


Figure 5. Drawing realized by a 4-year-old girl

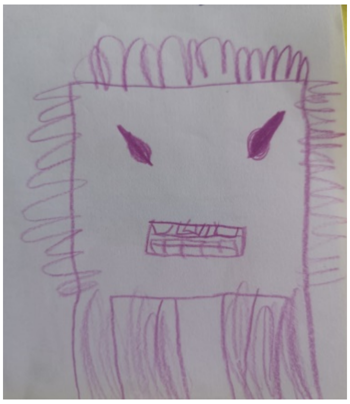


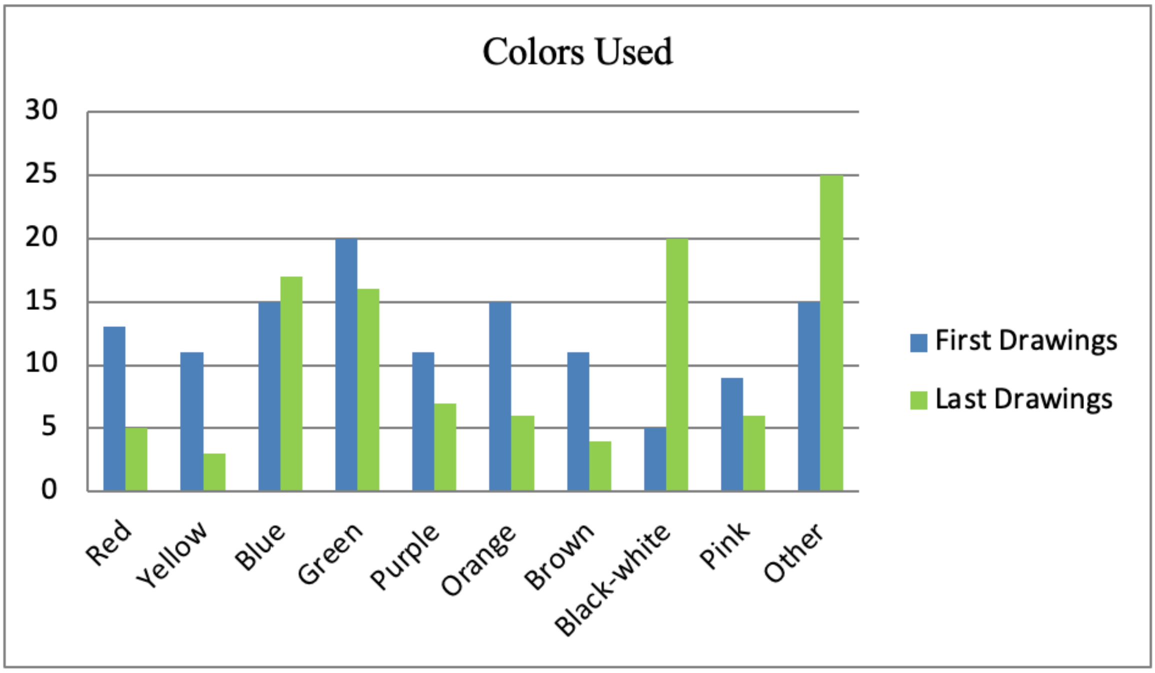
Figure 6. Drawing realized by a 5-year-old boy

*Figure 5-6 are placed to show which lines were used by children while drawing a robot.

3.1.3. Findings Related to Children's Drawings: Colors

When the colors used in the drawings were examined, it was seen in Table 4 that red, blue, green, purple, and orange colors were used frequently. It was seen that, in general, children used colors in their robot drawings.

Table 4. The distribution graphics of the **colors** used in children's first and last drawings



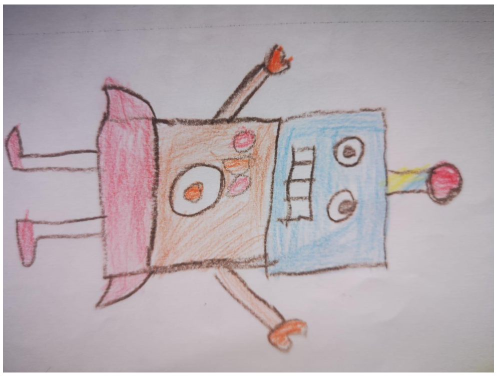


Figure 7. Drawing realized by a 6-year-old girl

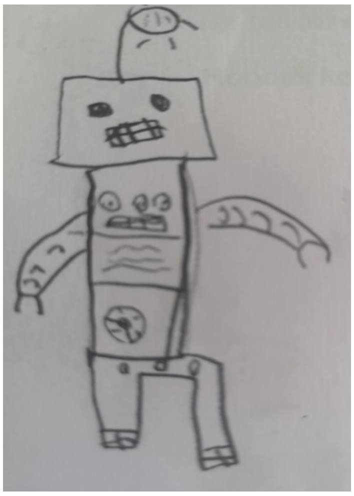


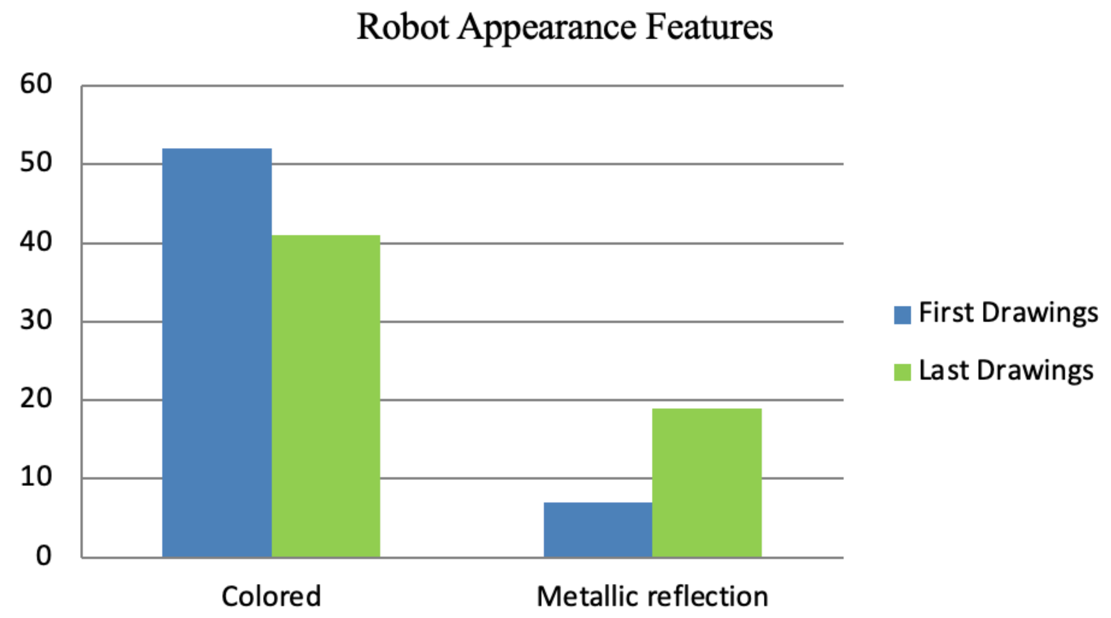
Figure 8. Drawing realized by a 6-year-old boy

*Figure 7-8 are placed to show which colors were used by children while drawing a robot.

3.1.4. Findings Related to Children's Drawings: Robot Appearance

In the first drawings, it was concluded that the number of drawings with metallic features was lower than the colored drawings as it can be seen in Table 5. In the last drawings, it was observed that the number of drawings with metallic features increased. The reason for this is thought to be the fact that the most functional one of the robots used as educational robots had metallic features. It can be interpreted that the robot named Evaluation attracts children's attention more and adopts more typical stereotypical features compared to other robots.

Table 5. The distribution graphics of the **robot appearance feature** in children's first and last drawings



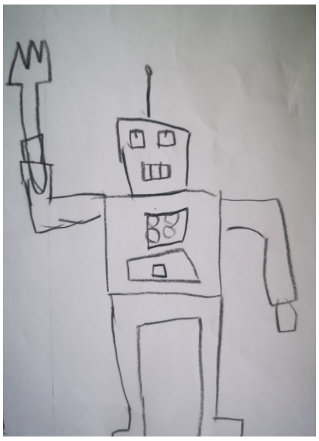


Figure 9. Drawing realized by a 5-year-old boy

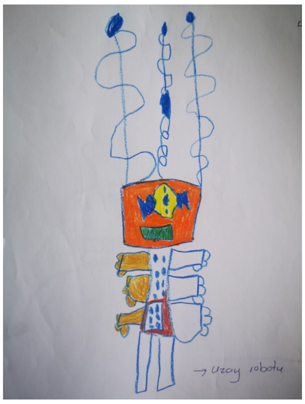


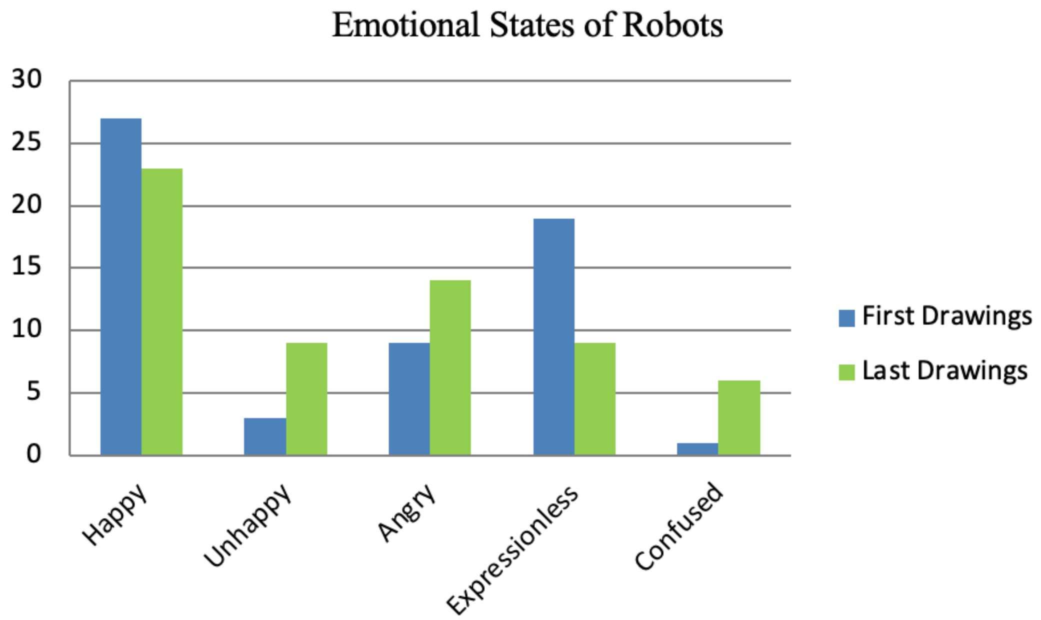
Figure 10. Drawing realized by a 5-year-old boy

*Figure 9-10 are placed to show how children expressed the robot appearance in their drawings.

3.1.5. Findings Related to Children's Drawings: Emotional State

In the robot drawings of children, emotional states of the robots were determined based on the facial expressions of the robots. With regards to the first drawings, it was concluded that happy and expressionless emotional states were in the majority as it can be seen in the Table 6. With regards to the last drawings, it was determined that the number of happy and expressionless drawings decreased while the number of angry, confused, and unhappy drawings increased.

Table 6. The distribution graphics of the **robot' emotional state** drawings in children's first and last drawings



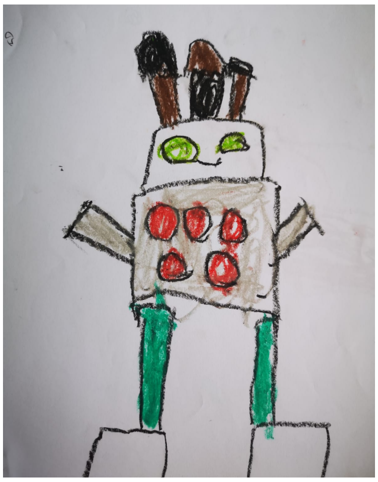


Figure 11. Drawing realized by a 6-year-old girl

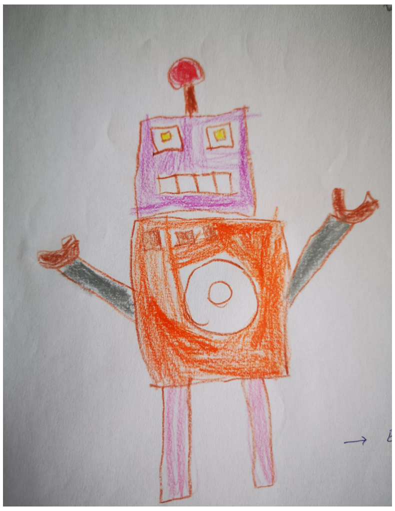


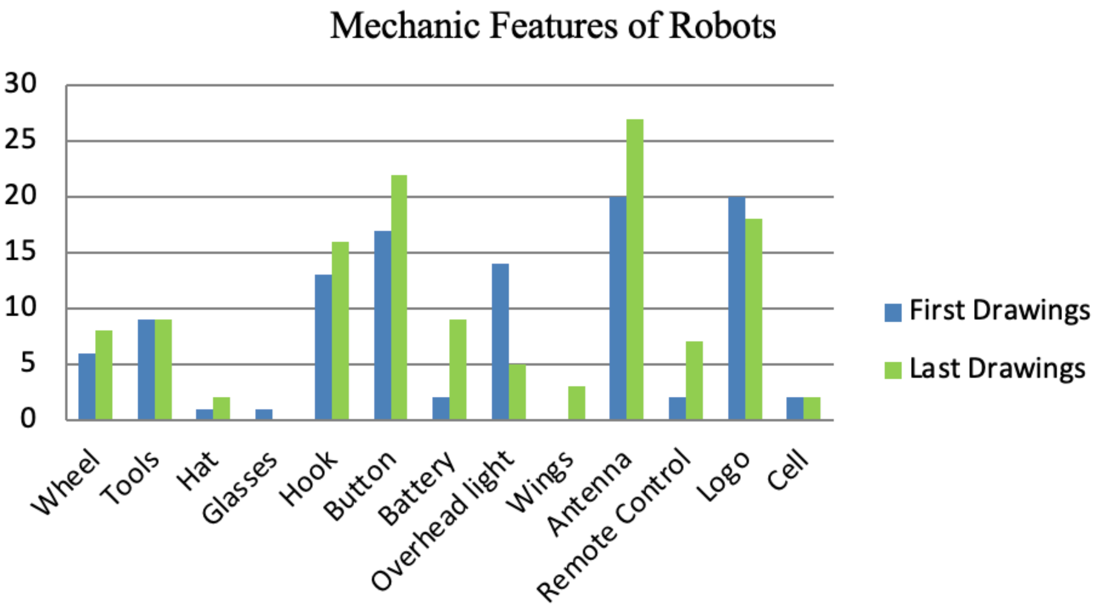
Figure 12. Drawing realized by a 5-year-old girl

*Figure 11-12 are placed to show how children expressed the robot's emotional state in their drawings.

3.1.6. Findings Related to Children's Drawings: Mechanic Features

When the mechanical features of the robots were examined, it was seen that there was an increase in the battery drawings in the last drawings compared to the first drawings. It was considered by the researchers that briefly mentioning the key information about how robots work during the education provided for children was effective in this case.

Table 7. The distribution graphics of the **mechanic features** of the robots in children’s first and last drawings



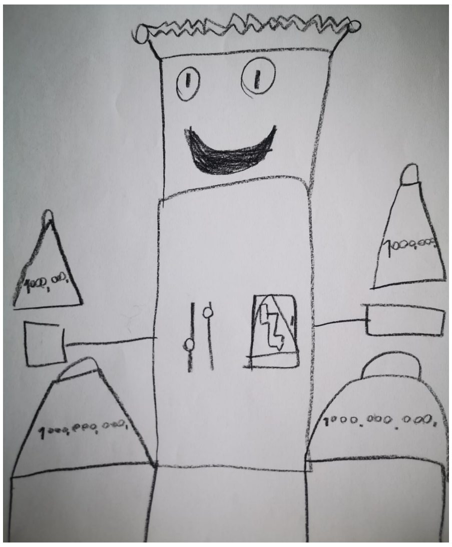


Figure 13. Drawing realized by a 6-year-old boy

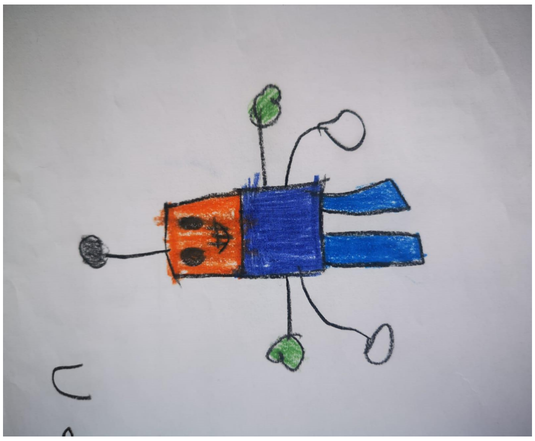


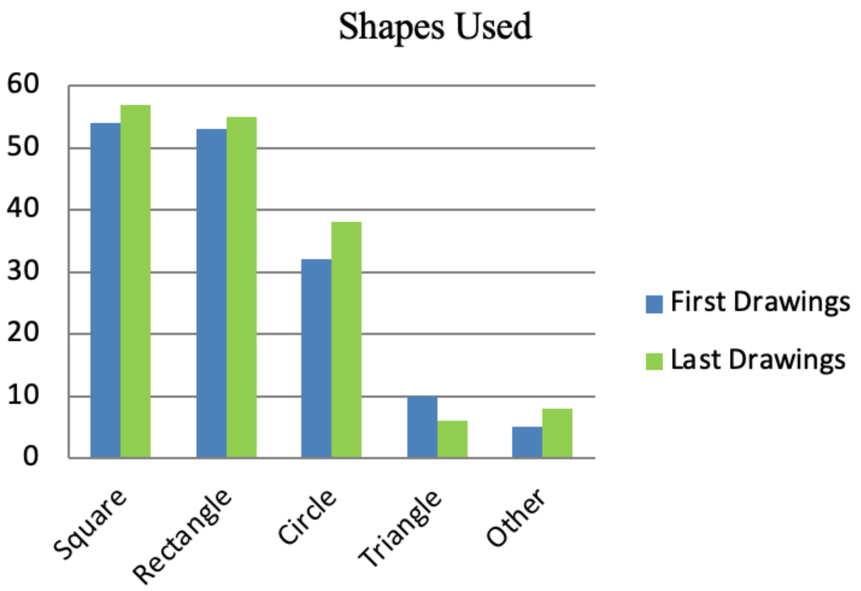
Figure 14. Drawing realized by a 5-year-old girl

*Figure 13-14 are placed to show which mechanic features were used by children while drawing a robot.

3.1.7. Findings Related to Children’s Drawings: Shapes

When the robot drawings of children were examined, it was striking that they mostly used square and rectangular shapes. There were no remarkable differences between the children’s first drawings before the educational activities and the last drawings.

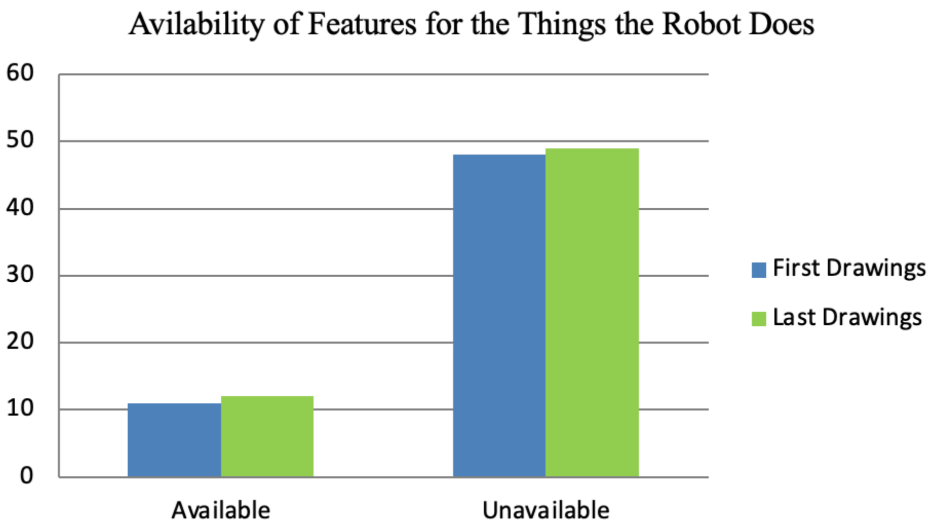
Table 8. The distribution graphics of the **shapes** children used in their first and last drawings



3.1.8. Findings Related to Children's Drawings: The Availability of Features for the Things the Robot Does

Considering the availability of features for the job the robot does, it was inferred that there were no remarkable differences between the children's first drawings and the last drawings as it can be seen in the Table 9. It was determined from the data that the robots mostly didn't have features for the things they did.

Table 9. The distribution graphics of **the availability of features for the things the robot does** in children's first and last drawings



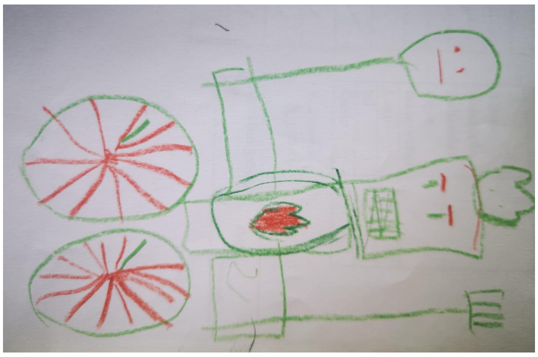


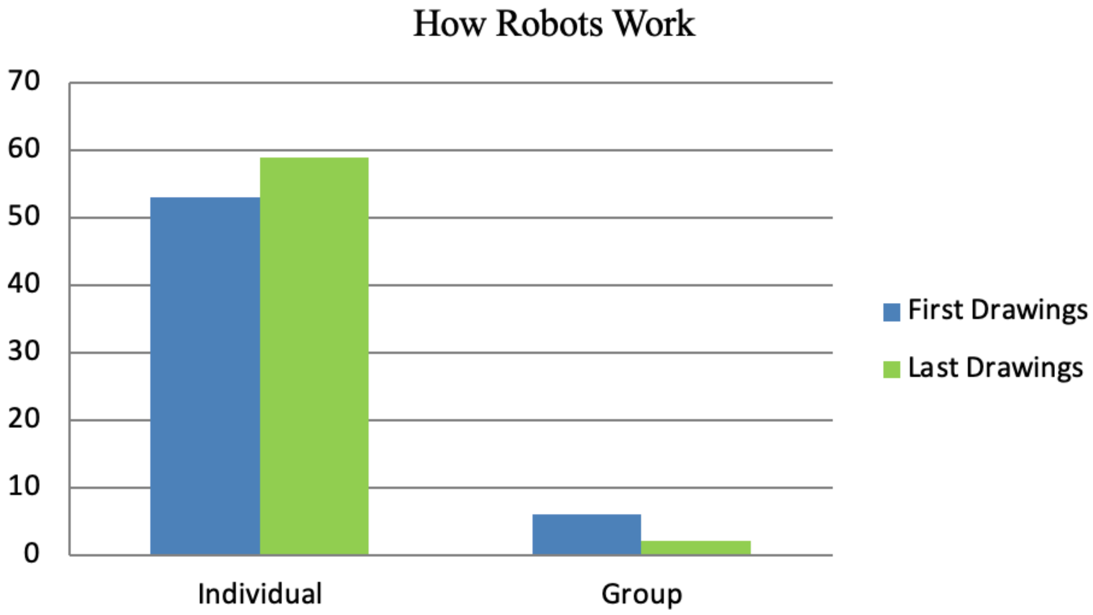
Figure 15. Drawing realized by a 6-year-old girl

*Figure 15 is placed to show the availability of features for the things the robot does in children's drawings.

3.1.9. Findings Related to Children's Drawings: Working Way

Considering the way robots work in children's drawings, it was determined that robots mostly worked individually. There was no remarkable difference between the children's first and last drawings as it can be seen in the Table 10. It was considered that children included individually working robots in their robot drawings because they were unable to experience robots working as a group.

Table 10. The distribution graphics of how **the way robots work** in children's first and last drawings



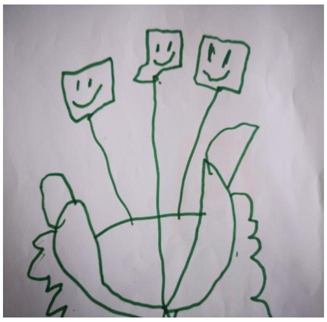


Figure 15. Drawing realized by a 4-year-old girl

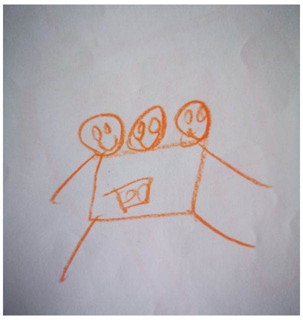


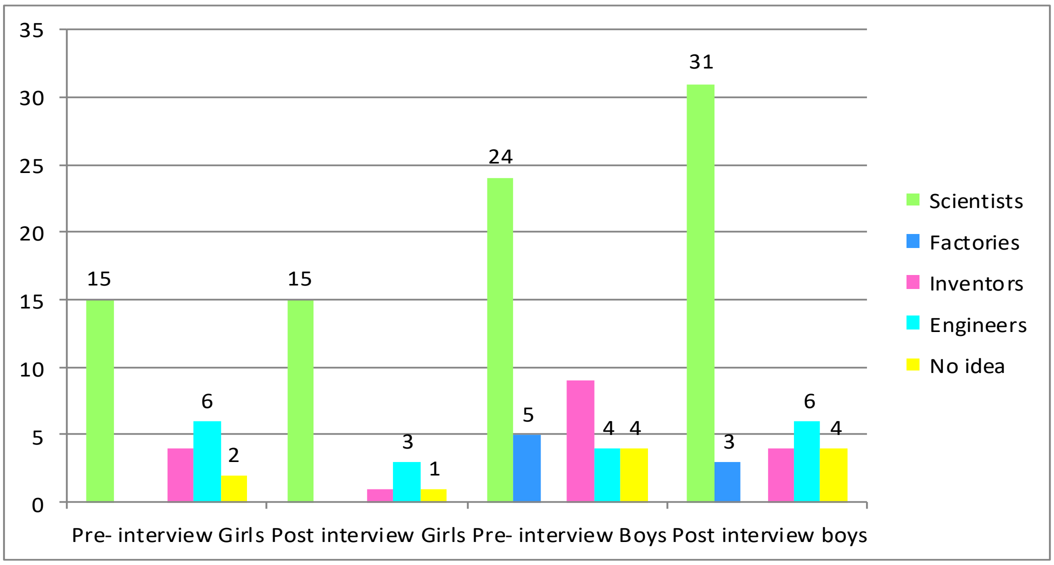
Figure 16. Drawing realized by a 5-year-old boy

*Figure 15-16 are placed to show how robots work in children’s drawings.

3.1.10. Findings Related to the Interviews with Children

The first finding is that children internalized the concept of scientist. In Turkish language there is a difference between scientist and man of a science. In the first interview children called scientist with their gender which is man of science. However, children internalized the scientist concept after this educational process. While this was not one of the aims of the research, it was one of the exciting results for the researchers. It was determined that the children responded to “Who builds the robots?” interview question. The answers of the children were examined under 4 headings which are scientist, factories, inventors, engineers, no idea. Intelligent people, hardworking people, professors’ answers are combined under the title of scientists. Robots, factories responses are combined under the heading of factories. Inventors, discoverers answers are combined under the title of inventors. The answers of masters, engineers, repairmen are combined under the title of engineers. This finding supports the assumption that the workshops held promoted the development of the concept of scientist in children.

Figure 17: The distribution graphics of the answer to the question *“Who builds the robots?”*



It was observed that the children participating in this study obtained information about how robots worked, who designed robots, how they were made, and the algorithmic and electronic system behind the operation of the robots thanks to this science festival. Another striking result obtained from the pre-test and post-test interviews was the information that most of the children had never seen a robot before. The quotations obtained from the responses of children were included in this study by coding C1, C2, C3, etc. Most of the children responded to the question of "Have you ever seen a robot?" as "No, I haven't seen" while one of the children provided the following response:

C3: "...I have seen it on TV..."

To conclude, most of the children stated that they saw a robot for the first time during the activities performed within the scope of the project.

Considering the question of "What is a robot?" in the first interviews.

C25 responded: "...It is something made of iron..."

C31 responded: "...It is a helper..."

C18 responded: "...It is something like a human..." Considering the same question in the last interviews.

C2 responded: "...Artificial intelligence..."

C34 responded: "...It is something that operates with electricity..."

C9 responded: "...A device made of electronic devices..."

When the responses given by the children for this question were examined, it was inferred that they had basic information on the term of artificial intelligence and the concept of electricity through which they got an idea of the operating principles of robots.

In addition to this, it was also determined that the children learned conductor and insulator terms, the distinction between these terms, and the concepts regarding the direction language such as "forward, backward, right, and left". The use of different learning methods and techniques with activities such as experiments, dance, STEM, art, plays, and family involvement in robotics coding workshop process is considered to improve high-level thinking skills such as problem-solving, predicting, reasoning, critical thinking, and finding creative solutions for the questions asked.

When the question of "Which activity did you like the most?" was asked to the children in the second interview, it was determined that they provided various responses. Some of the responses were as follows:

C5: "...I liked the Doc most..."

C1: "...BeeBot..." and

C43: "... The activity in which one of us described and the other drew..."

Based on this information, it can be inferred that performing activities with educational robots using various learning methods and techniques will provide permanence and significance in learning.

Considering the question of "Which activity did you like the most? Why?", some of the responses of children were as follows:

C3: "...I like it..."

C32: "... Because it is so entertaining..."

C60: "... It is so big, and this is the first time I saw it..."

C17: "... There is music coming out of it, interesting..."

The data obtained from the responses of children suggest that keeping children in the foreground of their permanent learning by having fun when planning activities supports this situation. It was observed by the researchers that the children attending the workshops increased their motivation in the process of getting robotics coding training and participating in the activities. Some of the children provided the following responses.

C2 responded: "... I learned right, left, back, and front terms..."

C45: "... I learned the directions..."

C63: "... I learned the directions and it was so good..."

Considering these responses of children, it can be inferred that concept of directions, position in space, and code blocks such as "move forward, turn right, and turn left" were developed by supporting them.

4. Discussion and Conclusion

In this study, the robot drawings and interviews of children continuing preschool education were examined. The collection of two different data on the same subject provided the researchers with the opportunity to obtain deeper information on the subject. It was concluded that children's last robot drawings had more mechanical features than their first drawings. There was a certain increase in the battery, wheel, button, and antenna in children's drawings. It was concluded that children used mostly angry, confused, and unhappy expressions in their robot drawings. When children's final drawings were examined, it was determined that there was an increase in the number of robots with mechanical reflection. As a result of the interviews held, it was concluded that the children internalized the concept of scientist and learned that robots were developed by scientists and engineers. In the first interviews, many children stated that they had never seen a robot while, in the last interviews, all children stated that they saw an educational robot. It is considered by the researchers that this early experience will increase the interests and motivations of the children in the education process. As a result of the interview, it was determined that the children enjoyed the process and internalized the concepts regarding position in space.

When examined in terms of developmental psychology, it is concluded that there are reasonable justifications for the use of educational robots in kindergartens. Piaget, one of the important names in the field of child development, emphasizes the necessity of presenting objects with different characteristics to children in early childhood. The process of using objects should continue without overstraining children, and learning should be planned step by step. (Piaget, 1962). Studies on the subject emphasize that even the process of turning the computer on and off will be a step (Alexander & Rackley, 2005).

For this reason, the main purpose of the study is to enable children living in disadvantaged areas to interact with educational robots. The second main reason is to examine children's perceptions of educational robots through pictures and interviews. When the obtained data are examined, it is concluded that all the children attribute human characteristics to the robots in both the first and the last robot drawings. They drew body parts such as eyes, hands, feet, and arms. In addition, they included the emotional states of the robots in their drawings. They include feelings like happy, unhappy, angry, and confused in their robot drawings. These results are similar to previous studies (Turkle, 2007; Sharkey & Sharkey, 2010). According to related research, it is emphasized that the reason why children attribute human characteristics to robots is the design of educational robots. In addition to this, it is shared that it would be appropriate to use transparent robot designs with visible mechanical structure as a suggestion (Boden, Bryson, Caldwell et all., 2017). However, another study has concluded that even after children learn that robots are programmable objects, they continue to think that robots have emotions and free will. In short, it is emphasized that there is no change in children's thoughts about educational robot’s feelings (Bumby & Dautenhahn, 1999).

5. Suggestions

Considering the results obtained from this study, the following recommendations can be made for practitioners, researchers, and policymakers: In the preschool period, longer-term training can be provided with educational robots, and activities can be organized. These activities will motivate children to discover and learn by offering a variety of materials and rich stimuli in education. Educational robots can be used from time to time in preschool educational activities to provide children with 21st-century skills. Providing coding education for children in the preschool period will support the training of young generations that not only consume technology but produce it. This study can be extended by applying it to preschool children in other public and private kindergartens for a longer period of time. Preschool teachers can carry out educational activities with robots for children and improve their coding skills through entertaining activities such as drama, play, movement, art, etc. Each child has a unique learning style and using different methods will ensure that learning becomes permanent. Instead of robot and coding tools with expensive technologies, multi-disciplinary education programs can be designed for preschool children with multifunctional materials with less cost. With these programs, children can develop creative thinking skills and their imagination, problem-solving skills, and the ability to solve a problem through different solutions. The extension of preschool coding education, which is included within the scope of compulsory education in some countries abroad, in Turkey, and carrying out intercultural studies on this subject can be encouraged.

References

Alexander C. & C. Rackley, (2005). Integrating Information Assurance (IA) Into K-5 Curriculum. *Proc. of the Information Security Curriculum Development Conference* ISBN: 1-59593-261-5/05/0009, 2005.

Barraza, L. (1999). Children’s drawings about the environment. *Environmental Education Research, 5* (1), 49-66

Beck, L. & Feldman, R. S. (1989). Enhancing children’s decoding of facial expression. *Journal of Nonverbal Behavior,* 13, 269-278.

Bers, M. U., Ponte, I., Juelich, K., Viera, A., & Schenker, J. (2002). Teachers as designers: Integrating robotics into early childhood education. *Information Technology in Childhood Education Annual,* 123-147)

Bers, M. U. (2008). *Blocks, robots and computers: Learning about technology in early childhood.* New York: Teacher’s College Press.

Bers, M., & Horn, M. (2010). *Tangible programming in early childhood: Revisiting developmental assumptions through new technologies*. In I. R. Berson & M. J. Berson (Eds.), High-tech tots: Childhood in a digital world (pp. 49–70). Greenwich: Information Age Publishing.

Boden, M., Bryson, J. Caldwell, D., Dautenhahn, K., Edwards, L., Kember, S. & Sorrell, T. (2017). Principles of robotics: Regulating robots in the real world. *Connective Science*, *29*(2), 124–129. DOI:https://doi.org/10.1080/09540091.2016. 1271400

Bornholt, L. J. & Ingram, A. (2001). Personal and social identity in children's self concepts about drawing. *Educational Psychology, 21*(2) 151-166.

Buldu, M. (2006). Young children's perceptions of scientists: A preliminary study. *Educational Research*, *48*(1), 121-132.

Bumby, K. & Dautenhahn. K. (1999). Investigating children's attitudes towards robots: A case study. *In Proceedings of the Third International Cognitive Technology Conference* (CT'99). Michigan: M.I.N.D. Lab, San Francisco, CA, 391–410. Retrieved from https://pdfs.semanticscholar.org/f6a5/9f92b02a956856964c5778f59f03fa4ab3ce.pdf.

Burkitt, E., Barrett, M. & Davis, A. (2009). Effects of different emotion terms on the size and colour of children's drawings. *International Journal of Art Therapy, 14*, 74-84.

Büyükkarabacak, O. (2008). *Çocuk Resimlerinde İmgelerin Yeri* (Yüksek lisans tezi). Marmara Üniversitesi, İstanbul.

Chang, N. (2012). What are the roles that children's drawings play in inquiry of science concepts? *Early Child Development and Care, 182*(5), 621-637.

Cox, M. V. & Moore, R. (1994). Children's depictions of different views of the human figure. *Educational Psychology: An International Journal of Experimental Educational Psychology, 14*(4), 427-436.

Creswell, J. W. (2013*). Beş Yaklaşıma Göre Nitel Araştırma ve Araştırma Deseni [Qualitative Inquiry & Research Design Choosing Among Five Approaches].* (Mesüt Bütün & Selçuk Beşir Demir, Trans. Eds.), Ankara: Siyasal Publishing.

Çakmak, A. & Darıca, N. (2012). 7-11 Yaş Grubu Kurumda ve Ailesi Yanında Büyüyen Kız ve Erkek Çocuklarının Anne Figürü Çizimlerinin Duygusal Gelişim Açısından İncelenmesi. *Elektronik Sosyal Bilimler Dergisi, 11*(41), 147-160.

Ehrlen, K. (2009). Drawings as representations of children's conceptions. *International Journal of Science Education, 31*(01), 41-57

Ersoy, A. & Türkkan, B. (2009). İlköğretim Öğrencilerinin Resimlerinde İnternet Algısı. *İlköğretim Online, 8*(1), 57-73, Retrieved from http://ilkogretim-online.org.tr

Güler, T. & Akman, B. (2006). 6 yaş çocuklarının bilim ve bilim insanı hakkındaki görüşleri. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi, 31* (31), 55-66. Retrieved from https://dergipark.org.tr/tr/pub/hunefd/issue/7807/102395

Gürtuna, S. (2004). *Çocuk ve Sanat Eğitimi.* İstanbul: Morpa Kültür Publishing

Ioannidou, A., Bennett, V., Repenning, A., Koh, K. H. & Basawapatna, A. (2011). Computational thinking patterns. Paper presented at *American Educational Research Association Annual Meeting* (AERA).

İskenderoğlu, L. (2006). *Çocuk Resimlerinde Görme Biçimleri* (Yüksek lisans tezi). Fırat Üniversitesi, Elazığ.

Johnson, J. H. (2003). Children, robotics, and education. *Artificial Life and Robotics*, *7*(1), 16-21.

Kasalak, İ. (2017). *Robotik Kodlama Eğitimlerinin Ortaokul Öğrencilerinin Kodlamaya İlişkin Öz Yeterlik Algılarına Etkisi ve Etkinliklere İlşikin Öğrenci Yaşantıları* (Yüksek Lisans Tezi), Hacettepe Üniversitesi, Ankara

Klein, C. A. (1982). Children's concept of the Earth and the Sun: A cross cultural study. *Science Education*, *65*(1), 95-107.

Lowenfeld, V. & Britain, L.W., *Creative and Mental Growıh,* 5th press, l1 Macrnillan Campany, New York, 1971

Malchiodi, C. A. (2005). *Expressive theories*. Guilford Publications

Metin, Ş. & Aral, N. (2012). Dört-Yedi Yaş Çocuklarının Resim Gelişim Özelliklerinin İncelenmesi. *Ankara Sağlık Bilimleri Dergisi, 1*(2), 55-69.

Odacı, M. M., & Uzun, E. (2017). Okul Öncesinde Kodlama Eğitimi ve Kullanılabilecek Araçlar Hakkında Bilişim Teknolojileri Öğretmenlerinin Görüşleri: Bir Durum Çalışması. *11. Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu* (702-709). Malatya: İnönü Üniversitesi.

Özdemir, M. (2010). Nitel veri analizi: Sosyal bilimlerde yöntembilim sorunsalı üzerine bir çalışma. Eskişehir Osmangazi Üniversitesi Sosyal Bilimler Dergisi, *11*(1), 323-343.

Patton, M. Q. (2014*). Nitel araştırma ve değerlendirme yöntemleri [Qualitative Research & Evaluation Methods].* (Mesut Bütün ve Selçuk Beşir Demir, Trans. eds.), Ankara: Pegem Publishing.

Pehlivan, H. (2008). Mythological drawings from Turkish elementary school children. *Elementary Education Online*, *7*(1), 150-156. [Online]: http://ilkö!retim-online.or g.tr

Piaget, J., (1962.), *Play, dreams, and imitation in childhood.* New York: Norton

Picker, S. H. & Berry, J. S. (2000). Investigating Pupil' Images of Mathematicians. *Educational Mathematics*,*43*(1), 65-94.

Prensky, M. (2001). Digital natives, digital immigrants. *On The Horizon, 9*(5), 1-6. Retrieved from https://www.marcprensky.com/writing/Prensky%20-%20Digital%20Natives,%20Digital%20Immigrants%20-%20Part1

Resnick, M. (2003). Playful learning and creative societies. *Education Update, 8*(6). Retrieved April 12, 2020, from https://web.media.mit.edu/~mres/papers/ education-update.pdf

Rijey, J. & Van Rooy, W. (2007). Perceptions about health in primary school children. *Teaching Science, 53*(4), 32-35.

Sadık, F., Çakan, H., & Artut, K. (2011). Çocuk Resimlerine Yansıyan Çevre Sorunlarının Sosyo-Ekonomik Farklılıklara Göre Analizi. *İlköğretim Online, 3*(10), 1066-1080.

Sayıl, M. (2004). Çocuk çizimlerinin klinik amaçlı kullanımı üzerine bir değerlendirme. *Türk Psikoloji Yazıları,7*(14), 1-13.

Schafer, S. (2012). Book Chapter: Optimizing Cognitive Coherence, Learning, and Psychological Healing with Drama-based Games. *Video Game Play and Consciousness.* Editor: Jayne Gackenbach. Nova Science Publishers.

Sharkey, A. & Sharkey. A.   (2010). The crying shame of robot nannies: An ethical appraisal. *Interactive Studies. 11*(2), 161–190. DOI:https://doi.org/10.1075/is.11.2.01sha

Stokrocki, M., & Kırışoğlu, O. T. (1996). *Ortaöğretim Sanat Öğretimi*. Ankara: YÖK Dünya Bankası Yayınları.

Tanyaş, B. (2014). An introduction to qualitative research methods: general 129 principles and applications in psychology. *Bulletin of Critical Psychology (Special Issue of Research and Method Discussions in Psychology),* 5, 25-38.

Turkle. S., (2007). Authenticity in the age of digital companions. *Interact. Studies, 8*(3) 501–517. DOI:https://doi.org/ 10.1075/is.8.3.11tur

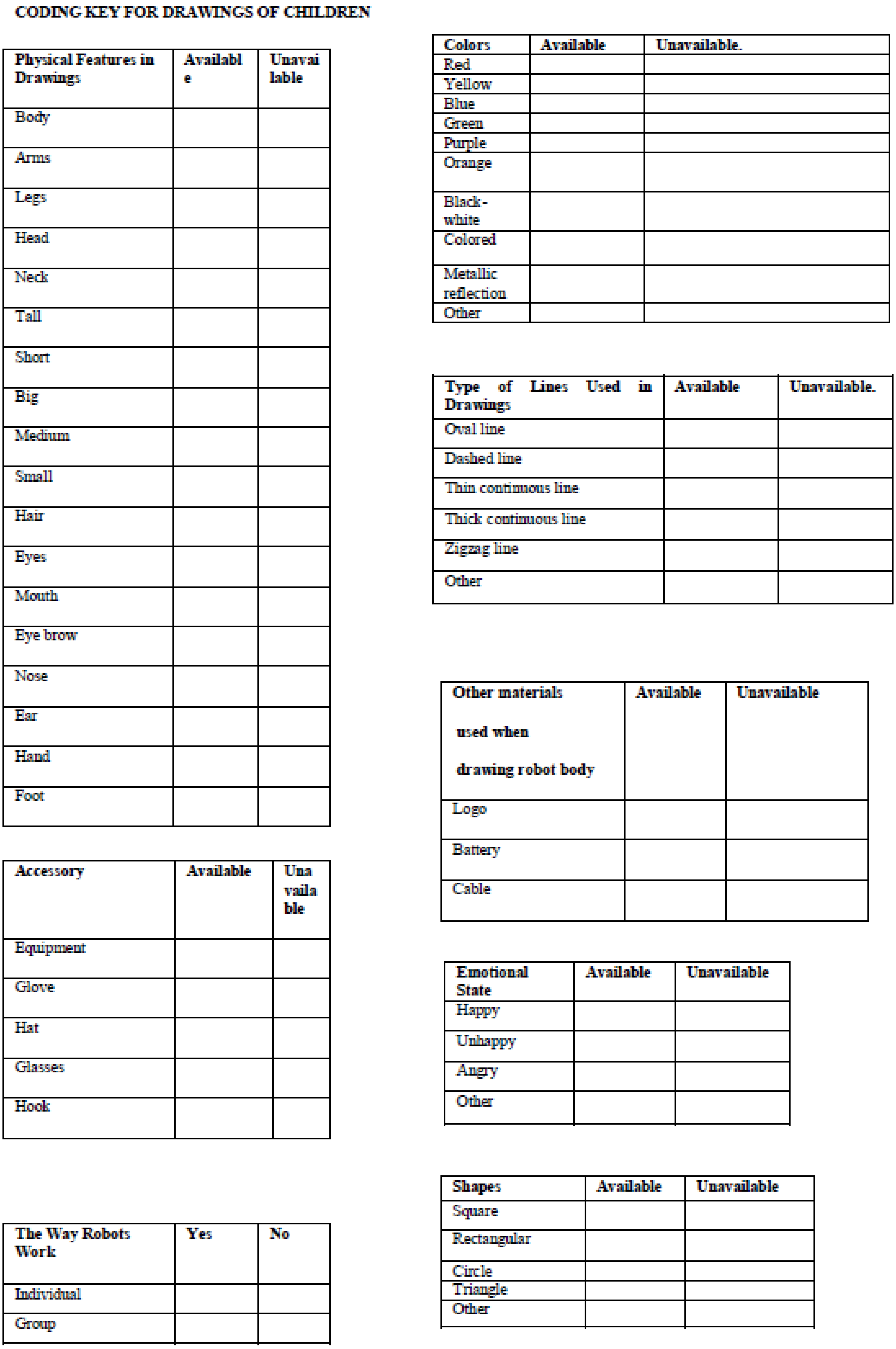
Türkkan, B., (2004), Okul öncesi dönem çocukların yaptıkları resimlerin aile yaşamlarına ilişkin ipuçları bakımından değerlendirilmesi. *I. Uluslararsı Okul Öncesi Eğitim Kongresi Bildiri Kitabı,* 79-97, İstanbul: YA-PA Publishing

Woods, S., Joerg, S. & J. Schulz (2004). The Design Space of Robots: Investigating Children's Views. Proceedings of the 2004 IEEE International Workshop on Robot and Human Interactive Communication Kurashiki, Okayama Japan Ek 20-22, retrieved from https://uhra.herts.ac.uk/bitstream/handle/2299/1895/902082.pdf;jsessionid=600469CC11A3BDDBF0C23F10058D10BE?sequence=1

Yavuzer, H. (2003). *Resimleriyle Çocuk; Resimleriyle Çocuğu Tanıma*. İstanbul: Remzi Kitabevi

Yıldırım, A. & Şimşek, H., (2016). *Sosyal Bilimlerde Nitel Araştırma Yöntemleri,* Ankara: Seçkin Publishing.

Appendix 1



CODING KEY FOR DRAWINGS OF CHILDREN

Physical Features in Drawings	Available	Unavailable
Body		
Arms		
Legs		
Head		
Neck		
Tall		
Short		
Big		
Medium		
Small		
Hair		
Eyes		
Mouth		
Eye brow		
Nose		
Ear		
Hand		
Foot		

Accessory	Available	Unavailable
Equipment		
Glove		
Hat		
Glasses		
Hook		

The Way Robots Work	Yes	No
Individual		
Group		

Colors	Available	Unavailable.
Red		
Yellow		
Blue		
Green		
Purple		
Orange		
Black-white		
Colored		
Metallic reflection		
Other		

Type of Lines Used in Drawings	Available	Unavailable.
Oval line		
Dashed line		
Thin continuous line		
Thick continuous line		
Zigzag line		
Other		

Other materials used when drawing robot body	Available	Unavailable
Logo		
Battery		
Cable		

Emotional State	Available	Unavailable
Happy		
Unhappy		
Angry		
Other		

Shapes	Available	Unavailable
Square		
Rectangular		
Circle		
Triangle		
Other		

www.ijcses.org