Thinking Skills and Creativity 38 (2020) 100714

Contents lists available at ScienceDirect

Thinking Skills and Creativity

journal homepage: www.elsevier.com/locate/tsc

The effect of robotics on six graders' academic achievement, computational thinking skills and conceptual knowledge levels

Serhat Bahadır Kert[b,*], Mehmet Fatih Erkoç[b], Sabiha Yeni[a]

[a] Leiden University, The Leiden Institute of Advanced Computer Science (LIACS), Leiden, Netherlands
[b] Yildiz Technical University, Department of Computer Education and Instructional Technology, Istanbul, Turkey

ARTICLE INFO

Keywords:
Programming education
Robotics
Block-based programming
Computational thinking
Conceptual knowledge level

ABSTRACT

This study compared the pedagogical effects of educational robot development and the block-based programming perspectives, which are used in programming education, on middle school students. Its participants were 78 sixth graders. Considering the students' preferences, 38 students were assigned to the experimental group, which studied with robotics (Lego EV3) sets, and 40 students were assigned to the control group, which studied with block-based programming environment (Scratch). All the topics of the programming unit, which are shown in the methods section, were taught to both groups for 10 weeks using the two different approaches. The change created by the implementation between the groups was tested for academic achievement, computational thinking skill efficacy perceptions, and conceptual knowledge levels. The results indicate that educational robotics develop middle school students' academic achievement and computational thinking skill efficacy perceptions more effectively than block-based programming environments. The connections between the concepts of the students who did robotics were also found to be more solid than those who worked with block-based software.

1. Introduction

Educational robotics has some characteristics that may directly or indirectly affect the pedagogical outcomes of different disciplines. Robots are physical objects that are designed and assembled. Their components include: tilt, temperature and position sensors, motors for movement, electric circuits, and a hub connector between the robot and the computer that controls it (Olabe, Olabe, Basogain, & Castaño, 2011). Robot production, addressing it as a development process that encompasses all these components, involves multifaceted thinking skills starting with the first goal-oriented analyses. Determining dimensions, basic movements, decision mechanisms, design and building stages, and finally programming the physical object require the combined use of fundamental skills such as algorithmic thinking, spatial memory and problem solving. Educational robotics thus offers students a constructive learning environment where they can seek solutions to real-world problems in interaction with their environment, and construct knowledge (Alimisis, 2013). One of the main goals of educational technology is to embody content. The use of concrete objects ensures that learning is based on constructive essentials (Piaget & Inhelder, 1969). The interaction of robots, designed physical objects with programmed behavior, with real life and the fact that the development outcomes can be observed harmonize with constructive learning theory.

* Corresponding author at: Yildiz Technical University, Faculty of Education, Department of Computer Education and Instructional Technology, C-213, Davutpasa Campus, Esenler, İstanbul, Turkey.
E-mail addresses: sbkert@yildiz.edu.tr (S.B. Kert), mferkoc@yildiz.edu.tr (M.F. Erkoç), s.yeni@liacs.leidenuniv.nl (S. Yeni).

https://doi.org/10.1016/j.tsc.2020.100714
Received 28 March 2020; Received in revised form 11 August 2020; Accepted 12 August 2020
Available online 22 August 2020

Knowledge construction emerges most effectively in the process of creating a real life product (Papert, 1980). Educational robotics offer students an entertaining and interesting technologically-integrated learning experience that involves both physical and mental activities (Eguchi, 2014). Robotics in a learning environment can be seen as having two different objectives: building robots and using robot development to teach concepts from other disciplines (Alimisis & Kynigos, 2009). As an instructional tool, robotics has an important effect on today's international trends in computer science. Resnick, Martin, Sargent, and Silverman (1996)) studied programmable cubes, a pioneering form of educational robotics, and found that they boost students' self-confidence as designers and shift their thinking towards computational ideas. In programming education, offering learners the opportunity to interact with concrete objects is the most important feature of robotics which distinguishes it from other code-writing applications. As in many different disciplines, abstract content may cause the learning objectives of programming education to become vague in learners' minds. Why are we learning this? is one of the most frequent questions asked by novice programming language learners. Learning objectives that lack effects in real life can negatively affect students' motivation to learn. Learners are glad to work with which they can influence their environments and on which they make a change using theoretical knowledge in code writing practices integrated into robot production. Such processes involve the movement of concrete objects, their position, and their interaction with our bodies in space (Hornecker & Buur, 2006). The thinking skills that educational robotics makes observable are computational thinking skills. Computational thinking means thinking how to state the solution to a problem in the form of a formula that can be processed by humans or machines (Wing, 2014), which can also be described as thinking like a computer scientist. ISTE-International Society for Technology in Education (ISTE, 2011) says that this skill domain has two components: characteristics and attitudes. Attitudes, such as direct outcomes of the learning process like creating formulas and abstraction, coping with uncertainty and teamwork skills, are the pedagogical goals of computational thinking skills education. Due to its extensive scope, there has been an increasing trend in the use of different instructional technologies in computational thinking education in recent years.

This study investigates the effect of educational robotics on middle school students' development to answer these research questions: For a student group who learned programming through educational robotics and another student group who learned programming through block-based programming environment, is there a significant difference between their:

- Self-efficacy perceptions of computational thinking skills (algorithm solving, problem solving, data processing and self-confidence)?
- Academic achievement in the programming unit?
- Conceptual knowledge levels about programming?

2. Theoretical framework

2.1. Educational robotics

Robotics indicates where embodiment in education is today. Robotics is an important field of professional education. It has become an educational technology with multidisciplinary applications. In educational robotics, becoming a partner in creation and coping with technical issues improve students' thinking skills, imagination, comprehension and teamwork. Not limited to the field of mathematics, this development can also be observed in social studies activities such as reanimating historical events, modeling interactions between social groups and resolving interpersonal issues (Ospennikova, Ershov, & Iljin, 2015). Ronsivalle, Boldi, Gusella, Inama, and Carta (2019)) found that robotics serves as a connection between learners and technology in difficult disciplines. They describe its stages as: needs analysis, separating goals, determining objectives, organizing the lab, identifying content, performance and evaluating products. Educational robotics that follows these stages with planned instructional design can positively affect learner development in different disciplines. Soliman (2019) monitored the development of a group of eighth-graders who did educational robotics during computer-aided educational activities. The study focused on creative thinking, and both the experimental and control groups included 15 students. The development of the creative thinking skills of the experimental group, which did robotics with Lego EV3 sets, was significantly higher than that of the control group. In their analyses of the importance of robotics in special education, Daniela and Lytras (2019) identified the important potential of robotics serving a supportive tool for learners experiencing problems in a specific field and for knowledge construction. Bargagna et al. (2019) did an eight-week study with children with Down's syndrome using bee-Bot robotic sets and found that educational robotics improved their interest, attention, level of interaction with adults and communication with peers. Teaching students how to take part in projects is one of the important outcomes of robotics. Cheng, Huang, and Huang (2013) investigated the effects of LEGO robot activities on students' achievement and in-group interaction and determined that they increased students' in-group interaction and positively affected their academic achievement.

The studies in the literature reveal the positive effect of educational robotics on learners. Comparing the effects of educational robotics and block-based software education on students' computational thinking skills development is an important aspect of this study.

2.2. Improving computational thinking skills

Computational thinking skills are among the fundamental twenty-first-century skills people should possess today (dos Santos Silva et al., 2018). Recently, there has been an increase in the number of studies on the development of computational thinking skills in the literature, not only with computers but also with a variety of physical tools. Wu (2018) examined the effect of board games on the development of computational thinking skills using these games: Robot Turtles, King of Pirates, Doggy Code, Robot Wars Coding

Strategy Board Game and Code Master. These games cost little, involve entertaining scenarios in programming and provide the game experience. They also have limitations such as not covering all the programming stages and being different from real programming. Applications with real programming environments should be used for the development of computational thinking. Many recent studies have attempted to improve computational thinking ability using Scratch for block-based programming (e.g., Vinayakumar, Soman, & Menon, 2018; Marcelino, Pessoa, Vieira, Salvador, & Mendes, 2018; Pérez-Marín, Hijón-Neira, Bacelo, & Pizarro, 2018; Jun, Han, & Kim, 2017; Rose, Habgood, & Jay, 2017). With different age groups, activities without computers and text-based programming languages are also used in education. Wong and Jiang (2018) studied the development of computational thinking skills through activities without computers and with the support of block-based software and found that the students improved. González, López, and Castro (2018) investigated the development of computational thinking skills with high school students using C++ programming in a text-based software environment and found significant positive outcomes.

Addressing the two different perspectives in the literature together, this study used block-based software programming and robotics with LEGO EV3 sets for the same set of learning objectives. The effect of the process on middle school students was evaluated in terms of programming academic achievement, computational thinking efficacy perceptions and conceptual development in robotics. The students' development in the computational thinking skills sub-factors of algorithm-solving efficacy, problem-solving efficacy, data-processing efficacy and self-confidence efficacy (Gülbahar, Kert, & Kalelioğlu, 2019) were investigated separately, and the links between them were analyzed. To investigate conceptual development, a word association network not found in the literature on computer science education was used, and the strength of the interaction between words was analyzed.

3. Method

3.1. Research design

This study used the pre-test post-test control group quasi-experimental design to determine the effect of different instructional technologies and teaching methods on six graders' programming and thinking skills. Students enrolled in two different courses, one of which is robotic programming and the other is block-based programming, set up by the school. Thus the students were separated into two groups in line their own preferences: a robotics group and a block-based programming group. For this reason, the method of this study is a quasi-experimental structure as students cannot be assigned to groups randomly. The same teacher worked with both groups for ten weeks to achieve the learning objectives, which were offered by the Ministry of Education (MEB, 2018). These learning objectives in the unit named "Programming" in the curriculum consist of; (1) the student explains fundamental concepts related to programming, (2) explains linear logic, (3) develops algorithms using linear logic, (4) explains the decision structure and its functions, (5) develops algorithms involving decision structures, (6) explains the loop structure and its functions, (7) develops algorithms involving loop structures, and (8) debugs by predicting the results of algorithms developed for different structures.

To achieve these objectives specified above, the experimental group students which studied with robotics with Lego Mindstorm EV3 sets, and the control group worked with Scratch, a block-based programming language. The study design is shown in Table 1.

3.2. Participants

The participants were 78 sixth graders who had never taken any block-based and robotic programming (11–12 year-olds). Considering the students' personal preferences, 38 of them were assigned to the experimental group, and the remaining 40 students were assigned to the control group. Due to students who did not participate in the pretest or posttest, the number of participants included in the analysis changed. During the research process, if the students did not take any pre-test or post-test, they were not included in the analysis. Since the data collection tools are applied to the students at different times, the students who do not take the tests also different. Since the data collection tools are analyzed and interpreted independently from each other, it is thought that students' differences will not have a negative effect on the scientific value of the study. The data of the 15 students who did not do the CTS before or after the implementation were not considered in the analysis, so the analysis of this data collection tool was done with 63 participants, 26 in the experimental group and 37 in the control group. The data of the 22 students who did not do the AAT at least once were not considered in the analysis, so the analysis of this data collection tool was done with 56 participants, 23 in the experimental group and 33 in the control group. Similarly, the data of the 8 students who did not do the WAT were not considered in the analysis.

Table 1
Design of the Study.

Group	Pretest	Instructional Strategy (10 weeks)	Posttest
Experimental Group	Academic achievement test (AAT)	Educational robotics	Academic achievement test (AAT)
Control Group	Self-efficacy perception scale for computational thinking skill (CTS) Word association test (WAT)	Block-based programming education	Self-efficacy perception scale for computational thinking skill (CTS) Word association test (WAT)

3.3. Data collection tools

The participants' pre-test and post-test computational thinking skills, academic achievement and conceptual knowledge levels were investigated separately. Computational thinking (CT) skills target behaviors and evaluation tools are shown in Table 2. Three different evaluation tools (the AAT, CTS, and WAT) were administered twice as pre-test and post-test. The AAT consists of questions about programming skills. The test was finalized by examining the item difficulty and item discrimination indices. The corrected item-total score correlation values of the factors in the scale used to measure the students' self-efficacy perceptions for CT skills ranged from 0.632 to .386, and the Cronbach's alpha coefficients ranged from 0.762 to .930 (Gülbahar, Kert & Kalelioglu, 2019). The Word Association Test (WAT) was used to determine the students' level of sense-making of CT concepts (loop, condition, command, etc.). The WAT is commonly used to determine students' cognitive structures, their ability to build links between mental concepts, conceptual changes and misconceptions (Gussarsky & Gorodetsky, 1988; Shavelson, 1974).

3.4. Data analysis

The participants' pre-test and post-test computational thinking skills, academic achievement and conceptual knowledge levels were investigated separately. Computational thinking (CT) skills target behaviors and

4. Results

4.1. Development of computational thinking skills

Multivariate analysis of variance (MANOVA) was used to determine whether there was a significant difference between the experimental and control groups' pretest CTS sub-scale scores (algorithm solving, problem solving, basic programming, data processing and self-confidence). Univariate and multivariate normality, a linear relationship between the dependent variables and the equality of variance-covariance matrices of the scores regarding the dependent variable, which are the major assumptions that should be ensured for the MANOVA test, were tested and confirmed. The results of the MANOVA test are shown in Table 3.

The MANOVA results indicate a significant difference between the control and the experimental groups' pretest sub-scores (Wilks' $\lambda = 0.395$, $F(5\text{-}57) = 17.48$, $p < .05$). According to the between-subjects effect table, shown in Table 4, significant differences were observed between the groups in all sub-scale scores except the problem-solving competence dimension. Considering the descriptive statistics regarding the data, the experimental group had higher mean scores on all the sub-scales. For this reason, the pretest data were used as the corrective (covariance) factor for the analysis of the posttest data of the CTS sub-scales.

The paired-samples *t*-test was used to determine whether there was a significant difference between the students' pretest and posttest CTS sub-scale scores. First, the assumptions of the *t*-test were met. Each factor of the scale and the total scores are shown in Table 5.

Table 5 shows that significant differences were found between the pretest and posttest scores. The sub-scales with significant differences were: problem solving ($t_{(22)} = 1.31$, $p > .05$) and data processing ($t_{(22)} = 1.56$, $p > .05$) for the experimental group; and problem solving ($t_{(22)} = 1.96$, $p > .05$), data processing ($t_{(22)} = 1.97$, $p > .05$), basic programming ($t_{(22)} = 1.97$, $p > .05$) and self-confidence ($t_{(22)} = 0.62$, $p > .05$) for the control group. Both the experimental ($t_{(22)} = 5.00$, $p < .05$) and control ($t_{(32)} = 4.69$, $p < .05$) groups' posttest CTS total scores differed significantly from their pretest scores. These results indicate that programing education

Table 2
CT Skills and Information on their Pedagogical Applications.

CT Skills	Learning Objectives	Data Collection Tools
Abstraction	The student can distinguish important from unnecessary information.	Academic achievement test (AAT)
Algorithmic Thinking	The student can recognize and apply the steps that should be followed to solve a problem.	
Pattern Recognition	The student can define similarities and differences in a problem to reach the solution.	
Logical Reasoning	The student can form a cause-and-effect relationship according to the data to reach the solution.	
Debugging	The student can detect and analyze the source of the problem.	
Formulization	The student can create the correct formula for the solution of the problem.	
CT Perspectives (Computational identity)	The student acquires the competences of designing algorithms, problem solving, data processing, basic programming and self-confidence.	CT self-efficacy perception scale (CTS)
CT Concepts (Conditions, loops etc.)	Students' level of sense-making of CT concepts and making links between these concepts increases.	Word association test (WAT)

Table 3
MANOVA Results by Group for the CTS Sub-scale Pretest Scores.

Source of Variance	Wilks' λ	F	sd	Error sd	p	η2
Group	.395	17.48	5	57	.000	.605

Table 4
Between-subjects effect table of the MANOVA test.

Variable	Dependent Variables	Sum of Squares	sd	Mean Squares	F	P
Group	Algorithm Solving	1301.98	1	1301.98	86.51	.000
	Problem Solving	42.35		42.35	2.76	.102
	Data Processing	119.13		119.13	9.79	.003
	Basic Programming	73.18		73.18	8.15	.006
	Self-Confidence	18.86		18.86	4.28	.043

Table 5
The Paired-Samples *t*-test Results for the Students' CTS Pretest and Posttest Scores by Group.

Group	Dependent Variable	Measurement	N	$\bar{x}$	S	sd	t	p
Experimental	Algorithm Solving	Pretest	26	20.42	3.85	25	5.54	.000
		Posttest		24.61	2.02			
	Problem Solving	Pretest	26	27.69	2.11	25	1.31	.203
		Posttest		28.42	1.84			
	Data Processing	Pretest	26	18.58	2.73	25	1.56	.131
		Posttest		19.38	1.63			
	Basic Programming	Pretest	26	11.00	2.48	25	3.59	.001
		Posttest		13.19	1.70			
	Self-confidence	Pretest	26	13.19	1.77	25	2.42	.023
		Posttest		14.15	1.22			
	Total Score	Pretest	26	90.89	7.91	25	5.00	.000
		Posttest		99.77	5.53			
Control	Algorithm Solving	Pretest	37	11.19	3.90	36	11.44	.000
		Posttest		20.62	3.80			
	Problem Solving	Pretest	37	26.03	4.79	36	1.96	.057
		Posttest		24.41	4.96			
	Data Processing	Pretest	37	15.79	3.93	36	1.97	.057
		Posttest		14.54	4.02			
	Basic Programming	Pretest	37	8.81	3.31	36	0.04	.967
		Posttest		8.84	3.08			
	Self-confidence	Pretest	37	12.08	2.30	36	0.62	.537
		Posttest		12.30	2.52			
	Total Score	Pretest	37	73.89	12.28	36	4.69	.000
		Posttest		80.70	12.47			

Table 6
The Corrected MANCOVA results for the Students' CTS Sub-scale Posttest Scores.

Source of Variance	Wilks' λ	F	sd	Error sd	p	η2
Group	.819	2.42	5	55	.047	.181

Table 7
Between-subjects effect table of the Corrected MANCOVA results for the Students' CTS Sub-scale Posttest Scores.

Source	Dependent variable	Sum of Squares	sd	Mean Square	F	P	**η2**
Group	Algorithm Design	75.93	1	75.93	9.97	.003	.025
	Problem Solving	44.51		44.51	3.53	.065	.070
	Data Processing	72.20		72.20	9.48	.003	.046
	Basic Programming	5.33		5.33	0.81	.373	.013
	Self-Efficacy	11.11		11.11	3.67	.060	.141

both block-based programming and educational robotics are effective ways of developing computational thinking skills.

Multivariate analysis of covariance (MANCOVA) was used to compare the experimental and control groups' CTS posttest scores.

Table 8
The *t*-test Results for the Students' AAT Pretest Scores by Group.

Group	N	$\bar{x}$	S	sd	t	p
Experimental	23	6.30	2.32	54	3.35	.001
Control	33	4.24	2.22			

The major assumptions of covariance analysis were analyzed and met. Both groups' computational thinking posttest scores are shown in Table 6.

Table 6 shows that the students' CTS sub-scale posttest scores, which were corrected according to their pretest scores, varied significantly in favor of the experimental group (Wilks' $\lambda = 0.819$, $F(5\text{-}55) = 2.42$, $p < .05$). To explain this result, the difference between the scores on the sub-dimensions of the scale and the effect of these dimensions on the scale total score were investigated, and the results are shown in Table 7.

Table 7 shows that the mean scores of the experimental group were higher than those of the control group on each sub-scale. According to the effect size values obtained for each sub-scale (η^2), the experiment has a large effect size. Significant differences were found in the sub-dimensions of algorithm design ($F = 9.97$, $p < .05$) and data processing ($F=9.48$, $p < .05$) in favor of the experimental group. These results indicate that programming education through educational robotics is a more effective method of developing computational thinking skills than block-based programming.

4.2. Development of academic achievement

This section discusses the results of the performance tests administered to the participants for the second research question. The unrelated samples *t*-test was used to determine whether there was a significant difference between the experimental and control groups' pretest AAT scores and to decide how to analyze the posttest. The normal distribution, unrelatedness of the sample and the equality of variances regarding the distribution of measurements, which are the major assumptions that are necessary to perform the *t*-test, were tested and met. The *t*-test results for the students' academic performance pretest scores by group are shown in Table 8.

The experimental and control groups' AAT pretest scores differed significantly ($t_{(54)} = 3.35$, $p < .05$), indicating that they were not equal in terms of academic performance prior to the experiment. For this reason, the pretest data as a corrective factor in the comparison of the groups' academic performance posttest data. The paired-samples *t*-test was used to determine whether there was a significant difference between the participants' pretest and posttest AAT scores. The paired-samples *t*-test results for the groups' pretest and posttest scores are shown in Table 9.

Table 9 shows that, for both the experimental ($t_{(22)} = 2.71$, $p < .05$) and control ($t_{(32)} = 2.48$, $p < .05$) groups, there was a significant difference between the pretest and posttest AAT scores in favor of the posttest. This indicates that both block-based programming and educational robotics are effective methods for programming education. Univariate analysis of covariance (ANCOVA) was used to compare the groups' posttest scores. Prior to the analysis, it was ensured that the equality of in-group regression tendencies, the linear relationship between the dependent variable and the covariance factor, and normal distribution, which are the major assumptions of the covariance test, were met. The covariance analysis results for the groups' posttest scores are shown in Table 10.

Table 10 shows the difference between the posttest scores, which were corrected according to the pretest, of the control and the experimental groups ($F_{(1\text{-}53)} = 6.32$, $p < .05$). This result indicates that robotics is a more effective way to improve sixth-graders' academic achievement in programming than block-based programming. The effect size (η^2) of the experiment is above medium, almost large, and 11 % of the posttest scores can be accounted for by the experiment.

4.3. Development of conceptual knowledge

A word association test (WAT) was used to investigate the students' conceptual development in programming. Stimulus words were presented to the students to determine the concept maps in their minds (Gussarsky & Gorodetsky, 1988), and the students were asked to note down the words that came to their minds related to the stimulus words within 4−5 min. The frequency table of the words the students used for each stimulus word is shown in Table 11 (Shavelson, 1974). The number of different answers for a word can be considered as an important and direct indicator of an individual's understanding of that word. Sense-making can be defined by the number and complexity of the links that students build between words (Schaefer, 1979).

Table 11 indicates a difference between the total number of words the experimental group students produced in favor of the posttest. Another important result was the decrease in the words, programming, coding, condition and loop, which are fundamental concepts of programming, in the control group students' posttest WAT results. On the other hand, a noteworthy improvement was observed regarding these concepts in the robotics group. This shows that the experimental group students created new concepts related to stimulus words and linked them with more words. Another method of analyzing WAT data is drawing concept maps using frequency tables. Concept maps represent the concepts students have in their minds for each stimulus word. The cut-off point technique was used to determine the response words' different frequencies (Bahar, Johnstone, & Sutcliffe, 1999). The cut-off points were frequencies of 5–8 (low), 9–12 (medium) and 13 or more (high), and separate concept maps were drawn for each cut-off point. Unlike the pretest, the control group gave responses in the high-frequency range to the stimulus words, algorithm and robot, on the posttest (Fig. 1).

Table 9
The Paired-Samples *t*-test Results for the Students' Pretest and Posttest AAT Scores by Group.

Group	Measurement	N	$\bar{x}$	S	sd	t	p
Experimental	Pretest	23	6.30	2.32	22	2.71	.013
	Posttest	23	7.70	2.18			
Control	Pretest	33	4.24	2.22	32	2.48	.018
	Posttest	33	5.55	2.32			

Table 10
The Corrected ANCOVA Results for the Groups' AAT Posttest scores.

Source of Variance	Sum of Squares	sd	Mean Square	F	p	η^2
Pretest	15.15	1	15.15			
Group	31.23	1	31.23	6.32	.015	.11
Error	261.90	53	4.94			
Total	2654.00	56				

Table 11
The Total Number of Response Words for Each Stimulus Word.

	Experimental		Control	
Stimulus words	Pretest	Posttest	Pretest	Posttest
Robot	24	29	22	27
Sensor	15	24	11	14
Programming	20	26	18	12
Algorithm	13	14	8	17
Coding	17	17	13	13
Condition	11	18	10	9
Loop	12	24	13	7
Command	14	23	15	16
Total	**126**	**175**	**110**	**115**

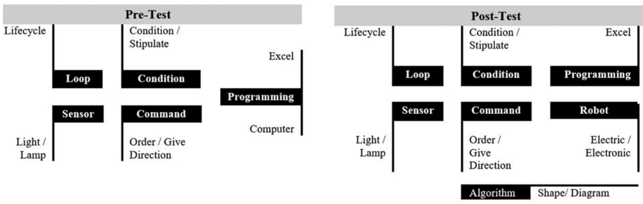

Fig. 1. The control group's pre-test and post-test high-frequency (13 or more) range WAT map.

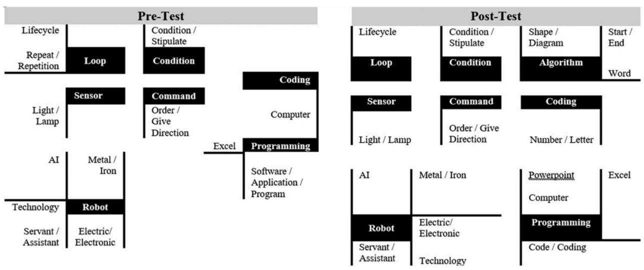

Fig. 2. The control group's pre-test and post-test medium-frequency (9-12) range WAT map.

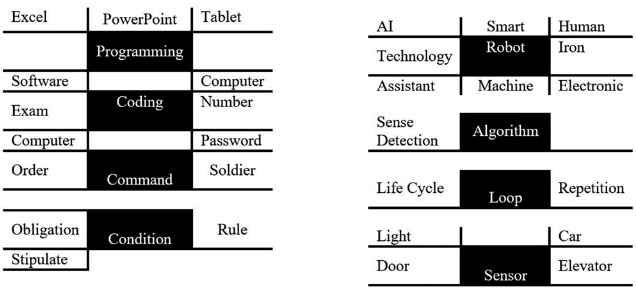

Fig. 3. The control group's pretest low-frequency (5-8) range WAT map.

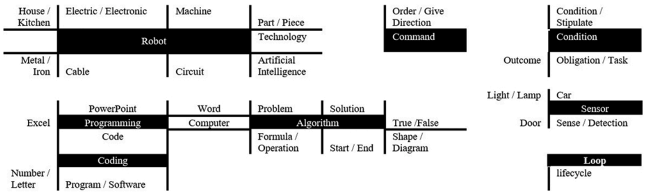

Fig. 4. The control group's posttest low-frequency (5-8) range WAT map.

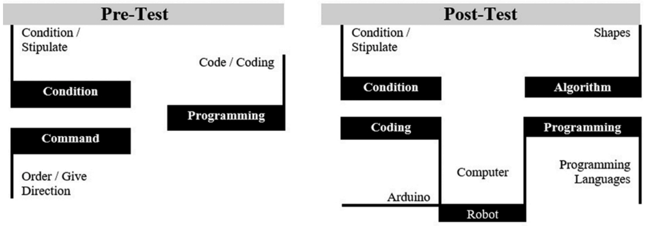

Fig. 5. The experimental group's pre-test and post-test high-frequency (13 or more) range WAT map.

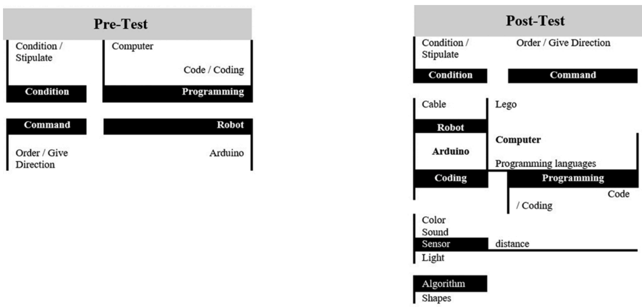

Fig. 6. The experimental group's pre-test and post-test medium-frequency (9-12) range WAT map.

The control group's pretest and posttest data for the medium frequency range indicated that similar concepts were included in the students' mental maps. (Fig. 2).

In the low-frequency range, the differentiation between the words was not at a high level, except for the algorithm stimulus word, while the response, sense/detection, were given on the pretest, the words, shape, diagram, start/end problem, solution, formula, true/false were given on the posttest. Except for the algorithm stimulus word, the experiment did not change the control group's cognitive comprehension of the stimulus words. The control group's low-frequency range pretest and posttest WAT maps are shown in Fig. 3 and Fig. 4.

The experimental group's pretest and posttest measurements indicated an increase in the word count in the high-frequency range after the experiment. While the high-frequency range synonymous words were given only when stimulus words were repeated (e.g., for the command stimulus word, the order response) on the pretest, more detailed responses to the stimulus words fell in the high frequency range (e.g., for the robot stimulus word, the Arduino response) after the experiment (Fig. 5).

Similarly, in the medium frequency range, there was a significant difference between the students' pretest and posttest concept maps. For example, responses to the sensor stimulus word did not fall in the medium frequency range on the pretest, but the responses, sound, distance, color and light, did fall in this frequency range on the posttest (Fig. 6).

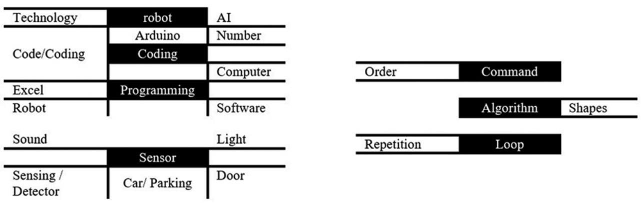

Fig. 7. The experimental group's pre-test low-frequency (5-8) range WAT map.

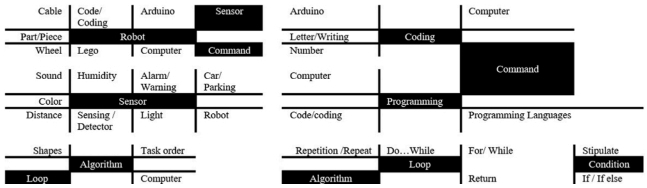

Fig. 8. The experimental group's post-test low-frequency (5-8) range WAT map.

In the low-frequency range, there were 5 connections between stimulus words on the pretest, but they increased to 14 on the posttest. This finding shows that robotics education taught the students to build more cognitive schemes around the stimulus words and to associate them with each other. The experimental group's low-frequency range pretest and posttest WAT maps are shown in Fig. 7 and Fig. 8.

Comparing the control and experimental group's posttest high-frequency range shows connections between concepts in the experimental group, but not in the control group. In the medium frequency range, the more detailed responses on the experimental group's posttest did not appear on the control group's posttest for some stimulus words. For example, while the light/lamp response for the sensor stimulus word was on the control group's posttest, the experimental group's responses also included light, color, sound and distance. Similarly, while the programming languages, C, Java and Python, were responses to the programming stimulus word on the experimental group's posttest, the control group's responses were more general words such as computer and code/coding. In the low-frequency range, one of the main differences between the experimental and control groups' posttests was the experimental group's significantly higher number of connections between concepts. Their interrelated network of concepts shows that they were able to access this information easily for problem solving, and that educational robotics had a positive effect on them.

5. Discussion and conclusion

This study's results obtained by the study indicate that both robotics and block-based programming positively affected the students' academic achievement, CT skill efficacy perceptions and conceptual knowledge levels about programming. Along with the recent trend of studies of computer science education without computers, this study corroborates the pedagogical efficiency of software-aided programming education. The academic achievement of the experimental group was significantly higher than that of the control group. The contribution of concrete objects to conceptual construction underlies this difference, as emphasized by Piaget and Inhelder (1969). In other words, the positive effect of embodying education was observed in robot development as an instruction tool. Cheng et al. (2013) reported the positive effect of Lego robot activities on students' academic achievement, which was also corroborated by this study's results. Considering the improvement in in-group CTS scores, this study's results are in line with those of Wong and Jiang (2018), and González et al. (2018). Soliman (2019) studied the effect of robotics on the development of creative thinking skills using Lego EV3 sets with results that are also similar to this study's results.

Both groups conceptual development in programming improved in terms of the total number of words generated. The control group students' posttest WAT results included fewer responses to the stimulus words, programming, coding, condition and loop, which are fundamental terms in programming, but a considerable improvement was observed regarding these terms in the experimental group, and their links between the terms also increased. The relevant studies have found that, when concepts in students' minds are in a branched and interrelated form, they more accessible and effective during problem solving (Kempa & Nicholls, 1983). In situations where concepts are linked weakly, it is not easy to get access terms using a different link, and problem solving that requires the link between the terms cannot be achieved. The cognitive structures in good problem solvers' minds are more complex and contain more links between concepts (Bahar et al., 1999). Thus, despite both groups' conceptual improvement in programming, the solidity of the experimental group's improved concept networks indicates a learning process that assists with knowledge construction, is meaningful and leads to more effective problem solving.

Both groups had significantly higher in-group CST total scores on the posttest, which indicates improved self-efficacy perceptions of their computational thinking skills. The control group did not significantly improve their posttest scores in the dimensions of basic programming efficacy perception ($t(22) = 0.04$, $p > .05$) and self-confidence efficacy perception ($t_{(22)} = 0.62$, $p > .05$), but the experimental group did. It is noteworthy that block-based programming supports students' academic achievement in programming, but fails to make a meaningful contribution to their self-efficacy perceptions, which lead to long-term learning acquisitions. The contribution of robotics, which yields concrete products, to the students' self-confidence is another striking result. This study's results indicate that educational robotics is a more effective method of developing computational thinking skills in programming education than block-based programming. This conclusion is consistent with those of other studies that have used different robotic tools such as those by Daniela and Lytras (2019); Bargagna et al. (2019), and Cheng et al. (2013). The literature includes many studies of the effect of block-based programming on students' computational thinking skills development such as Vinayakumar et al. (2018); Marcelino et al. (2018), and Pérez-Marín et al. (2018). However, this study found that educational robotics develops computational thinking skills more effectively than block-based programming, and this is an important contribution to the literature.

Addressing the pedagogical changes in programming education in three different dimensions is one of the points that contribute to the originality of this study. No studies in the literature examine the effect of programming on concept development through word association networks. This measurement technique, which is frequently used in linguistics, can be effectively used in computer science.

As this study shows that word association networks are data collection tools that contribute concepts to the perspective assessment approach proposed by Brennan and Resnick (2012) for evaluating the development of computational thinking skills.

6. Limitations

This study has the following limitations;

1 Selection bias may threat to the internal validity. Experimental and control groups were allocated according to the students' preferences. Students enrolled in a special robotics course, which was set up by school, were determined as the experimental group. We used a convenience sample of intact groups (classrooms) from a large urban school, we did not use random sampling. Pretest results indicate a significant difference between groups and the descriptive results show that the experimental group had higher mean scores for AAT and CTS. This effect on the analysis of the pretest data was controlled by using as the corrective (covariance) factor for the analysis of the posttest data.
2 Internal validity threats include testing effect; because students taking part in the research are tested more than once could influence their scores in the post-test. The pretests and posttests used in the study are identical. This effect was controlled in that for the academic achievement test, the order of questions was changed in the pretest and posttest to make it hard for students to remember questions. Also, tests are manifested equally in both experimental and control groups.
3 Instructional effect may be a threat to internal validity, depending on the teacher's comfort using technology. This effect was controlled by conducting research with the same teacher for both control and experimental groups. However, teacher can be in the different comfort level while using different technologies in two groups. Each type of technology affords opportunities for different actions and can help fulfill learning goals in different ways.
4 This experiment continued for 10 weeks to achieve the learning objectives. Regarding the internal validity of this study, history effect may have occurred if students have experienced additional learning (external to the experimental treatment) that affects their posttest scores. This effect was controlled in that teacher did not give any homework equally in both experimental and control groups. Also, this is controlled in that the general "external learning" events which may have contributed to the experimental group results may also contribute to the control group results.

7. Authorship statement

All persons who meet authorship criteria are listed as authors, and all authors certify that they have participated sufficiently in the work to take public responsibility for the content, including participation in the concept, design, analysis, writing, or revision of the manuscript. Furthermore, each author certifies that this material or similar material has not been and will not be submitted to or published in any other publication before its appearance in the *Journal of Thinking Skills and Creativity*.

Acknowledgements

All persons who have made substantial contributions to the work reported in the manuscript (e.g., technical help, writing and editing assistance, general support), but who do not meet the criteria for authorship, are named in the Acknowledgements and have given us their written permission to be named. If we have not included an Acknowledgements, then that indicates that we have not received substantial contributions from non-authors.

References

Alimisis, D. (2013). Educational robotics: Open questions and new challenges. *Themes in Science and Technology Education, 6*(1), 63–71.
Alimisis, D., & Kynigos, C. (2009). *Constructionism and robotics in education. Teacher education on robotic-enhanced constructivist pedagogical methods* (pp. 11–26).
Bahar, M., Johnstone, A. H., & Sutcliffe, R. G. (1999). Investigation of students' cognitive structure in elementary genetics through word association tests. *Journal of Biological Education, 33*, 134–141.
Bargagna, S., Castro, E., Cecchi, F., Cioni, G., Dario, P., Dell'Omo, M., … Sgandurra, G. (2019). Educational robotics in Down syndrome: A feasibility study. *Technology Knowledge and Learning, 24*(2), 315–323.
Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 Annual Meeting of the American Educational Research Association* (Vol. 1, p. 25). Canada: Vancouver.
Cheng, C. C., Huang, P. L., & Huang, K. H. (2013). Cooperative learning in Lego robotics projects: Exploring the impacts of group formation on interaction and achievement. *Journal of Networks, 8*(7), 1529.
Daniela, L., & Lytras, M. D. (2019). Educational robotics for inclusive education. *Technology Knowledge and Learning, 24*, 219–225.
dos Santos Silva, K., Odakura, V. V. V. A., & Pereira, N. P. (2018). Tools to support the teaching-learning of computational thinking in Brazil. October *2018 XIII Latin American Conference on Learning Technologies (LACLO)*, 284–291.
Eguchi, A. (2014). Educational robotics for promoting 21st century skills. *Journal of Automation Mobile Robotics & Intelligent Systems, 8*(1), 5–11.
González, F., López, C., & Castro, C. (2018). Development of computational thinking in High school students: A case study in Chile. November *2018 37th International Conference of the Chilean Computer Science Society (SCCC)*, 1–8.
Gülbahar, Y., Kert, S. B., & Kalelioğlu, F. (2019). Bilgi işlemsel düşünme becerisine yönelik öz yeterlik algısı ölçeği: Geçerlik ve güvenirlik çalışması. *Turkish Journal of Computer and Mathematics Education (TURCOMAT), 10*(1), 1–29.
Gussarsky, E., & Gorodetsky, M. (1988). On the chemical equilibrium concept: Constrained word associations and conception. *Journal of Research in Science Teaching, 25*, 319–333.
Hornecker, E., & Buur, J. (2006). *Getting a grip on tangible interaction: A framework on physical space and social interaction. Conference on Human Factors in Computing Systems CHI'06* (pp. 437–446). Montreal, Canada: ACM Press.

ISTE. (2011). *Operational Definition of Computational Thinking for K–12 Education*. Retrieved from http://www.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf.
Jun, S., Han, S., & Kim, S. (2017). Effect of design-based learning on improving computational thinking. *Behaviour & Information Technology, 36*(1), 43–53.
Kempa, R. F., & Nicholls, C. E. (1983). Problem-solving ability and cognitive structure-an exploratory investigation. *European Journal of Science Education, 5*(2), 171–184.
Marcelino, M. J., Pessoa, T., Vieira, C., Salvador, T., & Mendes, A. J. (2018). Learning Computational thinking and scratch at distance. *Computers in Human Behavior, 80*, 470–477.
MEB. (2018). *Bilişim Teknolojileri Ve Yazılım Dersi öğretim programı*. Ankara: Retrieved from http://mufredat.meb.gov.tr.
Olabe, J. C., Olabe, M. A., Basogain, X., & Castaño, C. (2011). *Programming and robotics with scratch in primary education. Education in a technological world: Communicating current and emerging research and technological efforts* (pp. 356–363).
Ospennikova, E., Ershov, M., & Iljin, I. (2015). Educational robotics as an inovative educational technology. *Procedia-Social and Behavioral Sciences, 214*, 18–26.
Papert, S. (1980). *Mindstorms: Computers, children and powerful ideas*. NY: Basic Books.
Pérez-Marín, D., Hijón-Neira, R., Bacelo, A., & Pizarro, C. (2018). Can computational thinking be improved by using a methodology based on metaphors and scratch to teach computer programming to children? *Computers in Human Behavior, 80*, 470–477.
Piaget, J., & Inhelder, B. (1969). *The psychology of the child*. New York: Basic Books.
Resnick, M., Martin, F., Sargent, R., & Silverman, B. (1996). Programmable bricks: Toys to think with. *IBM Systems Journal, 35*(3.4), 443–452.
Ronsivalle, G. B., Boldi, A., Gusella, V., Inama, C., & Carta, S. (2019). How to implement educational robotics' programs in italian schools: A brief guideline according to an instructional design point of view. *Technology Knowledge and Learning, 24*(2), 227–245.
Rose, S., Habgood, J., & Jay, T. (2017). An exploration of the role of visual programming tools in the development of young children's computational thinking. *Electronic Journal of E-Learning, 15*(4), 297–309.
Schaefer, G. (1979). Concept formation in biology: The concept' growth'. *European Journal of Science Education, 1*, 87–101.
Shavelson, R. J. (1974). Methods for examining representations of a subject matter structure in a student's memory. *Journal of Research in Science Teaching, 11*, 231–249.
Soliman, S. A. (2019). Efficiency of an Educational Robotic Computer-mediated Training Program for Developing Students' Creative Thinking Skills: An Experimental Study. *Arab World English Journal (AWEJ) Special Issue on CALL, 5*, 124–140.
Vinayakumar, R., Soman, K. P., & Menon, P. (2018). Fractal geometry: Enhancing computational thinking with MIT scratch. July *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 1–6.
Wing, J. M. (2014). Computational thinking benefits society. *40th Anniversary Blog of Social Issues in Computing, 26*.
Wong, G. K., & Jiang, S. (2018). Computational thinking education for children: Algorithmic thinking and debugging. December *2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, 328–334.
Wu, S. Y. (2018). The development and challenges of computational thinking board games. August *2018 1st International Cognitive Cities Conference (IC3)*, 129–131.

Serhat Bahadır Kert is associate professor and the chair of the department of computer education and instructional technologies (CEIT) at Yildiz Technical University. His research interests are computer science education for children, methodologies in programming language education, computational thinking, game-based education and development of e-learning environments. His PhD is in CEIT from Anadolu University. During his post-doctoral study at 2011–2012, he had a chance to study with Dr. Jianwei ZHANG who is known, in international scientific community, with his remarkable studies on pedagogical basis of knowledge building. That was an NSF project titled "Fostering Collective Progress in Online Discourse for Sustained Knowledge Building". He contributed to the process by preparing story boards of "Idea Thread Map (ITM)" which has been used for analyzing discourses on a timeline. He teaches course at both undergraduate and graduate levels and goes on to conduct national and international research on the pedagogical features of computer science education.