ICIC Express Letters
Part B: Applications
Volume **12**, Number **1**, January **2021**

ICIC International ©2021 ISSN 2185-2766
pp. **27–34**

ARDUCATION BOT: COMPUTATIONAL THINKING COURSEWARE WITH IOS MOBILE APPLICATION AND EDUCATIONAL ROBOTICS

Titipan Phetsrikran[1], Antony Harfield[1], Supiya Charoensiriwath[2] and Wansuree Massagram[1,*]

[1]Department of Computer Science and Information Technology
Faculty of Science
Naresuan University
Phitsanulok 65000, Thailand
{ titipanp60; antonyh }@nu.ac.th; *Corresponding author: wansureem@nu.ac.th

[2]National Electronics and Computer Technology Center
Patumtani 12120, Thailand
supiya.charoensiriwath@nectec.or.th

Received May 2020; accepted August 2020

Abstract. *The purpose of this project is to develop an educational platform for improving primary students' computational thinking ability. Arducation Bot combines tangible technology and mobile technology to create intuitively approachable teaching computational thinking. The result of the Arducation Bot was tested with 177 primary school students from Thailand. A clear pattern of improved computational thinking was demonstrated by the pre-test and post-test scores and related data from the Arducation Bot. This project presents a low-cost and intuitive teaching tool that can effectively develop skills in computational thinking and prepare students for computer science.*
Keywords: Computational thinking, Educational robotics, Tangible technology, Mobile technology

1. **Introduction.** The term "computational thinking" refers to a thought process to develop problem-solving skills that breaks down any task into smaller parts, finding patterns in each problem, and then logically presenting solutions using algorithms that can be repeated and followed. Computational thinking can help solve problems not only in computer science and mathematics but also in everyday life. Given its current relevance and importance, there is significant demand in Thailand and internationally for computational thinking in schools starting at a young age [1-3].

The purpose of this work is to provide a way to increase computational thinking skills by combining tangible and mobile technologies to develop a platform that is both accessible and effective. The iOS mobile application delivers challenging puzzles, which are divided into four units. Each unit teaches an important concept of computational thinking: sequences, loops, conditions, and conditions with loops. The Arduino-based robot car provides a tangible medium for interaction – young students can see the results of their thought and programming in the real world.

Tangible technology has proven to be an effective tool for children's mental development – mapping concrete objects to abstract reasoning such as computational thinking [4]. Existing computational thinking study curriculums such as code.org or MIT Media Lab's Scratch programming language often focus on creating codes on a personal computer or laptop. Using educational robotics such as Ozobot, Lego Mindstorms, or Arducation Bot offers the benefit of tangible technology and encourages increased collaboration among children. To date there have been two initial versions [5,6] of the Arducation Bot. The

DOI: 10.24507/icicelb.12.01.27

purpose of the current study is to further improve the system and its integrated courseware to teach the target computational thinking skills even more effectively and provide quantitative measurement of student improvement.

The organization of this paper is as follows. The background of related domains is described in Section 2. The methodology of how to construct the hardware, software, and courseware is illustrated in Section 3. Section 4 depicts the experimental procedures, their results, and the discussion of this research's significances. The conclusion and direction for future work for this study are summarized in Section 5.

2. **Literature Review.** The literature can be categorized into three main project-related headings: computational thinking, educational robotics, and courseware.

2.1. **Computational thinking.** The definition of computational thinking is a basic process for solving problems. There is a six-key concept of computational thinking, with five approaches [7,8]: (a) "logic" is reasoning that helps us explain why something happens, (b) "algorithms" are a sequence of instructions to solve problems, (c) "decomposition" is a process of breaking down a task into smaller pieces, (d) "patterns" are identifying details, creating rules and solving more general problems, (e) "abstracting" is simplifying or identifying something important without worrying about the details, and (f) "evaluation" is about making estimates of an objective in a systematic way. These concepts can improve the development of the five approaches: 1) "tinkering" is often to try something new to discover how it works, 2) "creating" is about making and planning something, 3) "debugging" is finding and fixing errors in code or algorithms, 4) "persevering" is a never give up attitude even though the problem is hard, and 5) "collaboration" is people working together to develop a good environment.

2.2. **Educational robotics.** The field of educational robotics includes many different facets, such as physical platforms, educational resources, and tangible technology. Robotics are commonly used in educational activities to transfer academic knowledge and skills related to Science, Technology, Engineering, and Mathematics (STEM) [9]. Educational robotics provides a tangible way that students can easily send instructions to a robot and have their input validation without yet having to learn syntax [10].

Active, cooperative, and problem-based learning using educational robotics and mobile technology are suitable for both undergraduate and graduate robotics education [11,12]. Another study shows the impact of educational robotics on children's technical, social, and science-related skills. The study depends on a two-point measurement (pre and post-test) to evaluate the impact, with each measurement a multiple-choice questionnaire [13].

Chang et al. believed educational robotics can help the kids in developing collaboration and communication, problem-solving abilities, critical thinking skills, and creativity among students [14]. Furthermore, teachers could completely include educational robotics in the young children's computer programming curriculum because the robots offer a better tinkering approach than computer monitors [15]. Thus, educational robotics are appropriate and have been practical to students of different age groups.

The arrival of educational robot use in schools has had a significant impact. For example, Nugent et al. used the robot together with geography technology to teach students about science, technology, engineering, and mathematics (STEM) [16]. Another study by Alimisis used educational robotics to identify new trends and challenges that focus on using robotics as a tool for creativity and other 21st century skills [17]. Williams et al. studied to estimate the impact of educational robotics on high school students' physics knowledge and scientific investigation skills [18]. Chin et al. used educational robotics to develop a system that provides an attractive teaching application about multimedia objects and its effect on student performance and motivation [19]. By no means educational robotics are the silver bullet. The tools are only as effective as the study plans and teaching materials.

2.3. **Courseware.** It is a term generally used to describe educational materials. These materials could be a kit to teach, train, or tutor the students. Most courseware is associated with technology-based materials. The term "courseware" is commonly referred to training for personal computers, software packages, or IT certification programs [20,21]. The followings are common courseware materials: instructor-led video or notes, self-directed computer-based training (CBT), interactive tutorials, and live or webinar.

The courseware in [22] demonstrates how STEM-driven computer science education supports the development of computational thinking at the high school. Well-known examples for computer science and computational thinking are code.org and ScratchEd [23,24] which use web delivered interfaces through which children learn how to write code. While these exercises offer a wide range of puzzles, the programming portion requires students to sit in front of their computers. This study wants to encourage the five computational thinking approaches: tinkering, creating, debugging, persevering, and collaboration. Playing with a tangible and mobile device in a group should encourage all five approaches, especially collaboration.

3. **Methodology.** The combination of tangible and mobile technologies created particularly for this project is called "Arducation Bot" [5,6]. Arducation Bot was designed to be an educational platform for improving primary school students' computational thinking ability. The initial version of the platform was found to have a few difficulties in both hardware and software. In hardware, the robot was not configurable due to all parameters being hard-coded and the line-following sensor did not track as well as needed. In software, the application UI was difficult to understand. This section describes the technical aspects of Arducation Bot and more generally, project implementation.

3.1. **Hardware.** The components of the robot include an Arduino UNO R3, three IR sensors, an ultrasonic sensor, an L298n motor driver, two DC motors, an HM-10 Bluetooth BLE module, and a 3.7V li-ion battery. Everything is placed on a plastic frame, with the motor shafts serving as axels for the wheels. The circuit diagram is shown in Figure 1. The Arduino provides control and processing power for the robot. The two types of sensors, ultrasonic and IR, provide input feedback. The motors, controlled by the L298, provide physical actuation. Bluetooth provides communication between the robot and the iOS application.

3.2. **Software.** Two kinds of software have been developed for Arducation Bot: Arduino and Swift. The two main functions are to control the robot and to communicate with an iPad. The Swift-based iOS application provides an interface between the user and the robot. The user's commands are sent from an iPad to the Arduino board through Bluetooth communication. The communication flow starts by the robot sending a "ready" message to the iPad indicating the robot's readiness to receive a command. The iPad sequentially sends commands to the robot, one at a time. After the robot finishes processing each individual command, it will send another "ready" message in order to receive the next command. Some commands are related to movement and others are related to obtaining information from the robot. For example, informational commands ask the robot if there is an obstacle in front of the robot. The puzzles and the interface in the iOS application styled after the curriculum and online sessions used by code.org. The target users of this platform are pupils above seven years of age. The UI/UX was designed to be intuitive and interactive, with users dragging and dropping the commands into the workspace as shown in Figure 2.

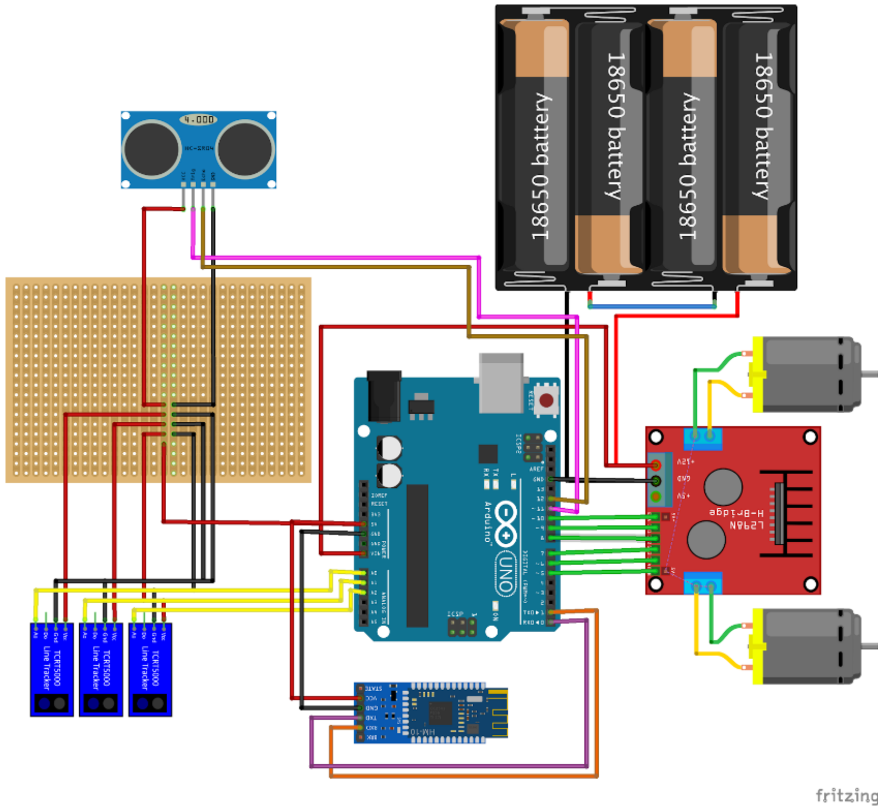

FIGURE 1. The Arducation Bot circuit diagram. Its components include an Arduino UNO R3, three IR sensors, an ultrasonic sensor, an L298n motor driver, two DC motors, and an HM-10 Bluetooth BLE module.

FIGURE 2. iOS application

The application starts the children off with a simple puzzle that explains basic commands to control the robot. The application contains twenty different puzzles, divided into four units of increasing complexity. The first unit requires the children to understand sequencing. The second unit teaches the concept of loops. The third unit is about conditionals. The fourth unit is a combination of a loop and a conditional. The design of the puzzles is explained in the courseware section below.

3.3. **Courseware.** Arducation Bot is modeled on concepts and designs shared by many online courses with block programming. The Arducation Bot platform consists of four units, each covering one topic with a series of target concepts, and the teaching of each target concept is built around one puzzle. The four units are Sequencing, Loops, Conditions, and Conditions with loops. In Unit 1 (Sequencing), children learn the concept of sequencing an activity into steps. In this unit, they also learn the basic commands and movements of the robot. In Unit 2 (Loops) children learn the concept of using a loop to repeat a procedure in order to accomplish a goal. In Unit 3 (Conditions) children learn the concept of expressing instructions based on a condition, using "If" and "Then". For example, "IF you meet an obstacle THEN turn left, but if you do not meet an obstacle, then continue straight". In this unit, the children learn how to divide a big problem into a sequence of smaller tasks. In the above example, there are three such smaller tasks: 1) check if there is an object, 2) if there is an object turn left, and 3) if there is no object, continue forward. Finally, in Unit 4 (Conditions with loops) children learn to combine the two concepts of If Conditions and Loops in order to reach a goal.

Children must learn various commands and think about the algorithm to solve these puzzles because there is a limit number of commands in each puzzle. For example, the first puzzle requires only two commands to solve. The loop concept is introduced with a puzzle that uses commands. Another concept that Arducation Bot uses to design and improve children's skills is the conditional puzzle. This puzzle uses the conditions command when meeting the obstacles then stop. Three examples of puzzles shown in Figure 3 illustrate the different themes. The design of all puzzles incrementally reveals the children's computational thinking concept as explained in the literature review section.

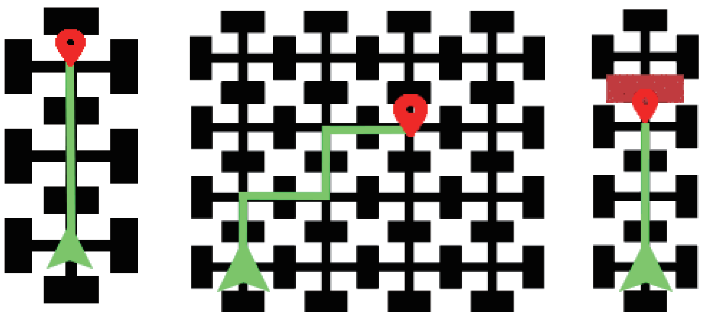

FIGURE 3. Examples of Arducation Bot: (left) the first puzzle, (middle) loop puzzle, and (right) condition puzzle

4. **Result and Discussion.** The Arducation Bot was tested in June 2019 with students from various primary schools (mostly around Phitsanulok, a provincial city in Thailand) during a one-day event called Computational Thinking for Kids, which was held four times, twice at Naresuan University and twice at St. Nicholas School in Phitsanulok. Each day, the students were split into ten groups, with four or five students in each group. Computational Thinking for Kids had two main sessions each day, called Unplugged (in the morning) and Arducation Bot (in the afternoon). At the beginning of the day, each student took a pre-test, and at the end of the day, they took a post-test. Data was collected from these two tests and processed to evaluate the difference in the computational thinking skills of each student before and after participating in the one-day event. The results are shown in Table 1.

This data was derived from a total of 177 students participating over the course of the four days. It is known that students at Events 2 and 3 had never studied computational thinking before, but Events 1 and 4 may have included some students who had studied

TABLE 1. Testing events and statistical analysis of the pre-test and post-test out of 10 points where $\bar{x}$ is the students' average score for each age group and σ is a standard deviation of the test scores

Event No.	**Participating students (Age)**	**Students count**	**Pre-test score**		**Post-test score**	
			$\bar{x}$	σ	$\bar{x}$	σ
1	Grade 1-6 (6-12)	46	5.64	2.72	7.25	2.16
2	Grade 2-3 (7-8)	50	4.7	1.19	6.95	1.82
3	Grade 4-5 (9-10)	50	3.62	1.83	5.98	2.54
4	Grade 1-6 (6-12)	31	5.48	2.82	7.6	1.69
		177	**4.86**	**2.14**	**6.95**	**2.05**

computational thinking previously. Looking at the mean points from the students' pre-tests and post-tests, the post-test scores were clearly improved, with a 43 percent increase over the pre-test scores. Standard deviations were 2.14 for the pre-test and 2.05 for the post-test.

The time required by each student to solve each puzzle was recorded in the iOS application. The puzzles are divided into four units: 1) Sequencing, 2) Loops, 3) Conditions, and 4) Conditions with loops. By solving these puzzles, the student should obtain computational thinking skills. The skills are in logic, decomposition, algorithms, abstraction, patterns, and evaluation. Each of the twenty puzzles requires the student to figure out one or more correct algorithms to move the robot from a starting point to a finishing point. When the student thinks they have figured out the correct algorithm(s) of the puzzle they are working on, they push the Run button. Then the robot will move according to their instructions (algorithms). However, if their algorithms are wrong, the student is informed and asked to try again. In order to understand how the Arducation Bot platform improves a student's computational thinking skills, the time required by each student to correctly answer each puzzle was automatically recorded. Figure 4 summarizes the average time (of all 177 students) spent correctly answering each of the 20 puzzles.

Interestingly, Units 1, 3, and 4 follow a similar pattern in Figure 4. In each of these three units, the student gradually used relatively more time per question at the start of the unit and relatively less time per question at the end of the unit. Only Unit 2

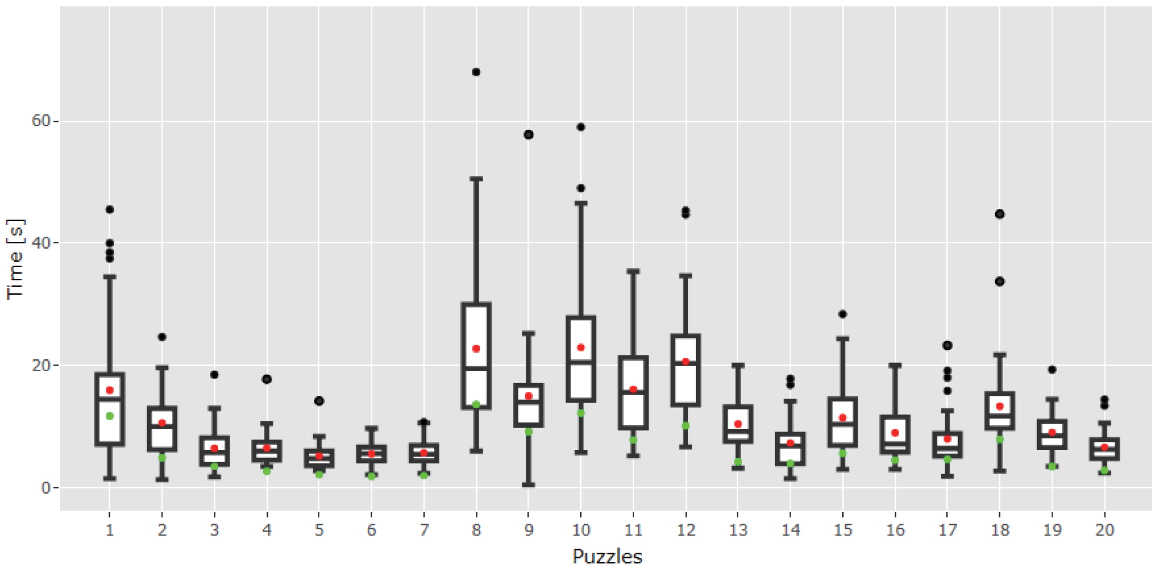

FIGURE 4. The average time required to successfully finish each puzzle

did not show this pattern. Since all the questions within one unit were approximately equally difficult, the shorter answering times as the unit progresses seems to show a clear improvement in the student's ability to understand the concepts in that unit. However, it is unclear why unit 2 did not follow the same pattern.

It should also be pointed out that conditions varied on four different test days. During the two test sessions at St. Nicholas School, students were on their home turf so to speak and being supervised by their regular teachers. Therefore, the students tend to be relatively well behaved. On the other hand, during the two other test sessions at Naresuan University, students were visiting the campus to attend and were surrounded by a new environment full of novel stimuli; thus they tended to be relatively less behaved and less focused on the task at hand. However, when the test results from the two locations are separated and compared, the results are strikingly similar to each other, as seen below in Figure 5. This similarity of outcome even in two different settings and atmospheres suggests that the test is valid and meaningful. The Arducation Bot platform successfully improved the students' computational thinking skills, enabling them to better understand and solve various computational tasks.

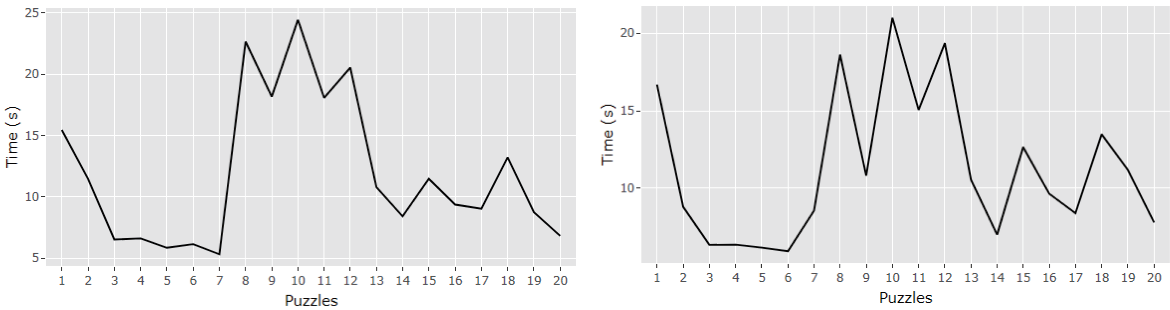

Figure 5. The average time required to successfully finish each puzzle at two different locations: (left) at the student's own school and (right) at Naresuan University

5. **Conclusions.** This study developed a tangible tool that utilizes mobile technology to create an educational platform in computational thinking. The results from 177 primary school students who participated in the Computational Thinking for Kids event have shown the potential of this courseware platform. A clear pattern of improved computational thinking was demonstrated by the pre-test and post-test scores and related data from the Arducation Bot. This platform presents a low-cost and intuitive teaching tool that can effectively develop skills in computational thinking and prepare students for computational thinking and computer science skills. To further prove the effectiveness of this proposed study, a comparison with extant pedagogical programs for computational thinking is needed. Furthermore, to broaden the impact of this study, the researchers are planning these two parallel efforts: the open-source hardware distribution for low-cost and the integration into classroom curriculums.

Acknowledgment. This study was made possible by a generous grant from the Thailand Graduate Institute of Science and Technology (TGIST) fund (Grant No. SCA-CO-2560-4530-TH). The authors would also like to thank the NU Maker Club and the Mobile Computing Laboratory at Naresuan University for their assistance.

REFERENCES

[1] E. M. Reeve, *Implementing Science, Technology, Mathematics and Engineering (STEM) Education in Thailand and in ASEAN*, Report Prepared for: The Institute for the Promotion of Teaching Science and Technology, 2013.

[2] Office of the Education Council, *The National Scheme of Education B.E. 2560-2579 (2017-2036)*, 1st Edition, Prigwhan Graphic, Bangkok, 2017.
[3] NECTEC, *KidBright*, https://www.nectec.or.th/news/news-pr-news/kidbright-coding.html/, Accessed on August 20, 2018.
[4] C. O'Malley and D. S. Fraser, Report 12: Literature review in learning with tangible technologies, *NESTA Futurelab*, 2004.
[5] T. Phetsrikran, W. Massagram and A. Harfield, First steps in teaching computational thinking through mobile technology and robotics, *Asian International Journal of Social Sciences*, p.37, 2014.
[6] T. Phetsrikran, W. Massagram, T. Phoka and A. Harfield, A feasibility study of Arducation Bot: An educational robotics and mobile application kit for computational thinking skills, *The 22nd International Computer Science and Engineering Conference (ICSEC)*, pp.1-6, 2018.
[7] Barefoot, *Computational Thinking*, https://www.barefootcomputing.org/concept-approaches/computational-thinking-concepts-and-approaches/, Accessed on September 18, 2017.
[8] BBC, *Introduction to Computational Thinking*, https://www.bbc.co.uk/bitesize/topics/z7tp34j, Accessed on September 19, 2017.
[9] S. Bargagna, E. Castro, F. Cecchi, G. Cioni, P. Dario, M. Dell'Omo and G. Sgandurra, Educational robotics in down syndrome: A feasibility study, *Technology, Knowledge and Learning*, pp.1-9, 2018.
[10] J. Shim, D. Kwon and W. Lee, The effects of a robot game environment on computer programming education for elementary school students, *IEEE Trans. Education*, vol.60, no.2, pp.164-172, 2017.
[11] G. Cielniak, N. Bellotto and T. Duckett, Integrating mobile robotics and vision with undergraduate computer science, *IEEE Trans. Education*, vol.56, no.1, pp.48-53, 2013.
[12] L. D. Riek, Embodied computation: An active-learning approach to mobile robotics education, *IEEE Trans. Education*, vol.56, no.1, pp.67-72, 2013.
[13] M. Kandlhofer and G. Steinbauer, Evaluating the impact of educational robotics on pupils' technical- and social-skills and science-related attitudes, *Robotics and Autonomous Systems*, vol.75, pp.679-685, 2016.
[14] C. W. Chang, J. H. Lee, P. Y. Chao, C. Y. Wang and G. D. Chen, Exploring the possibility of using humanoid robots as instructional tools for teaching a second language in primary school, *Journal of Educational Technology & Society*, vol.13, no.2, pp.13-24, 2010.
[15] M. U. Bers, L. Flannery, E. R. Kazakoff and A. Sullivan, Computational thinking and tinkering: Exploration of an early childhood robotics curriculum, *Computers & Education*, vol.72, pp.145-157, 2014.
[16] G. Nugent, B. Barker, N. Grandgenett and V. I. Adamchuk, Impact of robotics and geospatial technology interventions on youth STEM learning and attitudes, *Journal of Research on Technology in Education*, vol.42, no.4, pp.391-408, 2010.
[17] D. Alimisis, Educational robotics: Open questions and new challenges, *Themes in Science and Technology Education*, vol.6, no.1, pp.63-71, 2013.
[18] D. C. Williams, Y. Ma, L. Prejean, M. J. Ford and G. Lai, Acquisition of physics content knowledge and scientific inquiry skills in a robotics summer camp, *Journal of research on Technology in Education*, vol.40, no.2, pp.201-216, 2007.
[19] K. Y. Chin, Z. W. Hong and Y. L. Chen, Impact of using an educational robot-based learning system on students' motivation in elementary education, *IEEE Trans. Learning Technologies*, vol.7, no.4, pp.333-345, 2014.
[20] Idea Group. Lnc., *What is Courseware*, https://www.igi-global.com/dictionary/disruptive-technologies-and-education/6098/, Accessed on August 21, 2018.
[21] Margaret Rouse, *Courseware*, https://whatis.techtarget.com/definition/courseware/, Accessed on August 21, 2018.
[22] R. Burbaitė, V. Drąsutė and V. Štuikys, Integration of computational thinking skills in STEM-driven computer science education, *Global Engineering Education Conference (EDUCON)*, pp.1824-1832, 2018.
[23] F. Kalelioğlu, A new way of teaching programming skills to K-12 students: Code.org, *Computers in Human Behavior*, vol.52, pp.200-210, 2015.
[24] K. Brennan, *ScratchEd*, https://scratched.gse.harvard.edu/, Accessed on October 28, 2019.