Tactile Scratch Electronic Block System: Expanding Opportunities for Younger Children to Learn Programming

Yunju Jo, Seok-Ju Chun, and Jungwoo Ryoo

***Abstract*—This paper introduces our work on the development of a novel system for applying MIT's Scratch to teaching classes of four to eight-years-old students. Scratch is a visual, block-based programming language designed for anybody to create a computer program without the worry of syntax errors by assembling icon-like command blocks. However, four to eight-year-old students have trouble using a computer mouse or keyboard and face difficulties when trying Scratch programming. This research developed a tactile, electronic block system that allows students to manipulate physical objects in a tangible way to conduct their programming tasks. The system consists of a Scratch simulator and physical, electronic blocks embodying the Scratch user interface shapes. We taught programming to the classes of second-grade elementary school students (eight-years-old) using our system. The results are encouraging. Our subjects' interest in programming improved from 3.23 to 4.0 out of the scale of 5, and fifteen students out of twenty five were able to solve nine questions on sequence, loop, and control structure successfully, which are fundamental concepts of programming.**

***Index Terms*—Scratch programming, tangible block programming, electronic block system, programming education, early elementary students.**

I. Introduction

Computational Thinking (CT) as a necessary ability in a modern society is gaining traction. In education sectors, there are many research projects for developing computational thinking among students. Among these efforts, programming is one of the best ways to help people develop their CT skills. There are various block-based coding languages designed for educational purposes. Among these, Scratch is popular and available in more than 40 languages and 150 countries. The Lifelong Kindergarten Group at the Massachusetts Institute of Technology (MIT) Media Lab runs the Scratch program, and Millions of people of all ages use it [1]. The core goal of Scratch is to teach programming to people without any prior experience. Scratch allows students to focus on programming concepts and logic instead of grammatical errors. It also makes programming more approachable [2].

However, according to the Scratch statistics page (https://scratch. mit.edu/statistics/) that shows the age distribution of new Scratchers, the number of four-to-eight-year-old users is substantially lower than that of twelve-year-olds, whose number is the biggest. That is, the twelve-year-old new Scratchers are 5,292,649 while four and six-year-olds are 104,693 and 234,751 respectively. Compared to twelve-year-olds, their percentages are 2.0 and 4.4 percents. It is well known that students have difficulties in using software like Scratch because they lack efficient hand-eye coordination (necessary for using a mouse) [3]. Especially, children under eight had trouble dragging and dropping [4].

To address this problem, we developed a tactile Scratch electronic block system that allows children to touch physical blocks and do Scratch programming. Our system consists of Scratch electronic blocks and a Scratch simulator. Using our solution, We conducted classes for early elementary school students. The results are encouraging. Their interest scores improved from 3.23 to 4. More than the fifty four percent of students correctly answered their questions in our assessment of basic programming concepts. We expect to four-to-eight-year olds to learn Scratch programming more easily by taking advantage of our system.

II. Related Works

As Computer Science is becoming important, there are many efforts made to promote Computational Thinking (CT) among children. The CS Framework [5] provides concrete concepts and methods for K-12 CT education. There are various ways to develop CT, but programming is one of the most effective options for teaching children CT. Elementary schools often use visual block-based programming languages to help their students develop their CT skills [6], [7].

Visual block-based programming languages, such as Scratch[1], Snap[2], Blocky[3], Raptor[4], and Makecode[5] have been developed to reduce the complexity of programming for inexperienced users and novice programmers. Among them, Scratch is most widely used in schools around the world as a means to introduce basic computer programming to children. Scratch is a free educational programming language which is geared towards kids ages of 8-16 from 2nd grade to high school with over 57 million registered users and 58 million shared projects.

Some block-based programming languages provide a scalability to handle sensors and actuators as real-world hardware. Scratch 3.0, the latest version of Scratch, also provides an extension to handle various physical devices

Manuscript received December 12, 2020; revised February 26, 2021.

Yunju Jo is with Sin-mook Elementary School, Republic of Korea (e-mail: yunju0514@gmail.com).

Seok-Ju Chun is with Seoul National University of Education, Republic of Korea (corresponding author; e-mail:chunsj@snue.ac.kr).

Jungwoo Ryoo is with Penn State University, United States (e-mail: jryoo@psu.edu)..

[1] https://scratch.mit.edu
[2] https://snap.berkeley.edu
[3] https://developers.google.com/blockly
[4] https://raptor.martincarlisle.com
[5] https://makecode.microbit.org

doi: 10.18178/ijiet.2021.11.7.1529

such as Makey Makey[6], Micro:bit[7], LEGO MINDSTORMS EV3[8], and etc., that is, it provides collections of extra blocks to program these physical devices.

Recently, several studies have been conducted on tangible block programming. Nicolas Villar *et al*. [8] developed a Torino which is a physical programming language for teaching computational learning to blind and low vision children. Using Torino, children can create code while connecting physical instruction pods and tuning the parameter dials to create music, audio stories, or poetry. Conference'17, July 2017, Washington, DC, USA Anon.

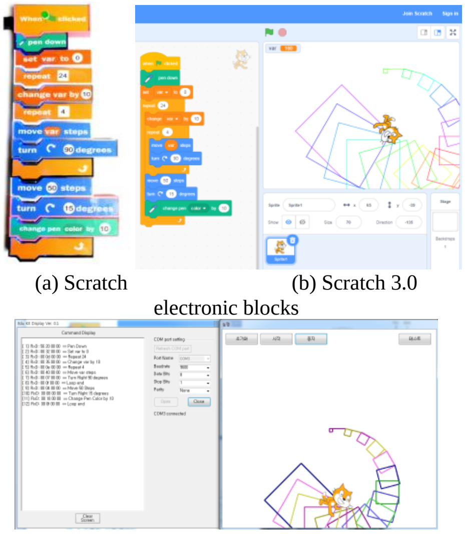
Zhiyi Rong *et al*. [9] developed a tangible programming toolkit for engaging blind and visually impaired students to learn fundamental programming concepts by creating a simple melody. This toolkit contains a set of blocks comprising tangible syllable blocks and several distinctive function blocks representing different programming concepts. They aim to help students to learn and get used to computing programming.

Márcia Alves *et al*. [10] proposed a tangible block programming system which is referred to as Tactode. Tactode can create a puzzle with tangible pieces, take a photo and upload it in the app. In the application, the puzzle is compiled into executable code for robot, and it is possible to execute a real robot such as Ozobot, Cozmo, Sphero or Robobo robots etc.

III. Scratch Electronic Block System

The Scratch electronic system consists of physical blocks (Fig. 1(a)) and a simulator (Fig. 1(c)). We designed our blocks to mimic the Scratch blocks (Fig. 1(b)) provided by MIT Scratch programming in terms of their shape and functionality. Our electronic Scratch block solution allows users to connect blocks with their hands just like LEGO blocks instead of dragging and dropping virtual blocks in a Personal Computer (PC)-based Scratch programming environment using a mouse. The blocks are magnetic and connects to each other easily and similar to their virtual counterparts in functionality. Our Scratch blocks are either masters (comparable to event blocks) or slaves. After connecting a master block to several slave blocks, a user can push a green flag button and trigger the master to communicate with the slaves and read the overall block structure. The master block then sends commands to the simulator which implements a stage where Sprites move around .

Fig. 2(a) shows how our master electronic block Micro-controller Unit (MCU) [11] communicates with its slave block MCUs. The General Purpose Output (GPO) of a master block sends a signal to the General-Purpose Input (GPI) [12] of a slave below. Through Tx (Transmit), the slave block sends its information in the form of a signal to the master via Rx (Receive), and vice versa. The master block sends a sequence of command block information (i.e., block program) to a Scratch simulator, resulting in the movements of Sprites. General-Purpose Input (GPI) of a slave below. Through Tx (Transmit), the slave block sends its information in the form of a signal to the master via Rx (Receive), and vice versa. The master block sends a sequence of command block information (i.e., block program) to a Scratch simulator, resulting in the movements of Sprites. Fig. 2(b) shows the connection of a master block and 6 slave blocks.

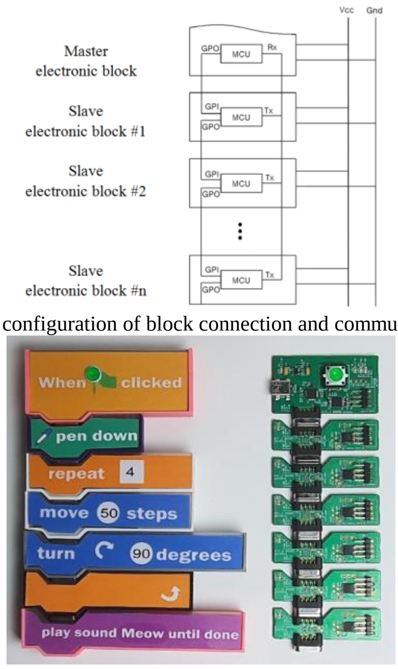


(a) Scratch (b) Scratch 3.0

electronic blocks

(c) Scratch simulation

Fig. 1. Scratch electronic block system.



(a) The configuration of block connection and communication

(b) A connected state of 1 master block and 6 slave blocks

Fig. 2. The interconnection of a master and slave blocks.

This research adopted 23 electronic blocks for teaching second graders Scratch programming in an elementary school. To do this, we first analyzed the CS Framework's K-12 standard [5]. We chose the Algorithm and Programming Concept as our core for lessons out of the five concepts available through the CS Framework [5]. We then selected the 'Great programs that include sequences, simple loops, and conditionals' in the Grade 3-5 (Age 8-11) standard of the concept. Considering the second grader level in elementary education, we finalize the choice of the Scratch 3.0 blocks for teaching sequences, simple loops, and

[6] https://makeymakey.com

[7] https://microbit.org

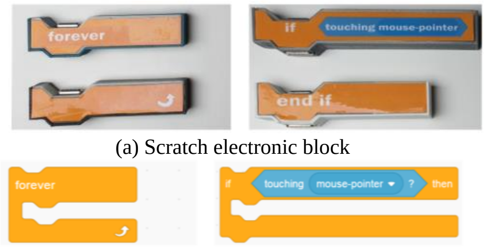
[8] https://www.lego.com/en-us/product/lego-mindstorms-ev3-31313

conditionals. Table I lists the details of the Scratch electronic blocks we used.

TABLE I: SCRATCH ELECTRONIC BLOCKS

Category	Implemented Electronic blocks
Events	When flag clicked
Control	Forever Repeat 4 Repeat 24 If then If then else Wait 1 sec Wait 2 sec
Motion	Move 50 steps Move 100 steps Turn right 15 degrees Turn right 90 degrees Go to random position Turn right(90, 180, 270) degrees (using a rotary sensor)
Sound	Play sound Meow Play sound record
Sensing	Touch mouse-pointer
Variable	Set var to 0 Set var to 1 Change var by 1 Change var by 10
Pen	Pen down Pen up

The Scratch electronic blocks implement most of the Scratch 3.0 blocks. However, there are some differences depending on the shape of the blocks. When a Scratch block represents a pair of commands (e.g., looping commands like forever and repeat or conditional commands like if then and if then else), we implemented them as separate blocks as in Fig. 3. The differences are minor and doesn't cause any issues when students use the blocks to do Scratch programming.



(a) Scratch electronic block

(b) Blocks with equivalent functions

Fig. 3. Differences in Scratch 3.0 and the electronic blocks.

IV. EXPERIMENTS

Our research subjects are twenty five second grade elementary students (eight-year-old) in Seoul, South Korea. The duration of our course was between June 25, 2020 through July 30, 2020. We met every Thursday for forty minutes during their creative experiential learning activity time. Two of the twenty five students who participated in our class had prior experience in block coding. The rest of the students (23) were first-timers.

The research question is 'Will the classes using tactile Scratch electronic blocks affect interest in programming?'. The hypothesis is 'Students will have higher interest in programming after class than before class.'

Interest in programming means more attention, concentration, positive feelings or relatively enduring predisposition toward programming activities [13].

A. Scratch Programming Classes

We developed a Scratch course to test our hypothesis. We designed our course content to meet the CS Framework standard and used our Scratch electronic blocks developed inhouse. We had total six classes and students learn sequences, loops, and conditionals during the classes. Table 2 shows the class sequence of our curriculum. At the first class, students had pre-test of interest of programing and at the last class, they had post-test of interest of programing and interview.

TABLE II: SCRATCH PROGRAMMING CLASS COURSE

Period	Contents	Concepts
1	-Introduction -Pre-test of Interest in Programming	
2	-Introduction of 'Scratch' -Observing blocks and block shapes -Studying 'sequence' concept - Moving a cat sprite 50 steps and 100 steps -Drawing a 100-step-line and playing meow sound	Sequences
3	-Understanding '90°' -Studying 'loop' concept -Using 'Repeat4' and 'Repeat24' blocks - Drawing a 200-step-line using a 'repeat' block -Drawing one step using a 'repeat' block	Sequences Loops
4	-Studying 'event' concept -Moving a cat sprite 50 steps when the flag clicked -Playing a record sound if a cat touch a mouse pointer -Making algorithms and expecting results	Sequences Loops Conditionals
5	-Reviewing -Creating my own artwork -Presenting and Sharing the artwork	Sequences Loops Conditionals
6	-Pre-test of Interest in Programming -Test of programming basic concept -Interview	

When using their Scratch electronic blocks, students wore vinyl gloves for the sake of safety(due to COVID-19)as shown in Fig. 4. and assembled electronic blocks and checked the simulator results individually. We shared the video of assembling blocks to implement algorithms with students.

Fig. 4. Scratch electronic blocks use in the class.

B. Results

We assessed the level of interest in programming after teaching a course using our Scratch electronic blocks to students who didn't have any prior experience with block coding.

We used a five-level Likert scale for students to express interests by providing them with survey questions. Table 3 shows the composition of interest survey in programming.

TABLE III: COMPOSITION OF INTEREST SURVEY IN PROGRAMMING

Category	Number of Questions
Interest toward programming	Q1, Q2
Interest toward programming learning	Q3, Q4
Interest toward programming activities	Q5
Willingness to continue programming class	Q6
Interest toward programming related careers	Q7, Q8
Noanxiety about programming class	Q9, Q10

The survey questions for each category include:

Q1: I am knowledgeable about programming.
Q3: I look forward to programming classes.
Q5: I like programming activities.
Q7: I am interested in programming professions.
Q9: I am not afraid of programming course content.

To test the effect of our course, we analyzed the changes before and after the delivery of our curriculum. Table 4 shows the result. The p value is 0.000 and less than 0.05. Therefore, there is a significant statistical difference.

TABLE IV: PRE-POST TEST RESULTS OF INTEREST IN PROGRAMMING

	M	D	T	P
Pre-class	3.23	0.701	-5.178	0.000**
Post-class	4.00	0.555		

Students also submitted the reflections on their use of Scratch electronic blocks, which also indicated their heightened curiosity and interest in Scratch programming. Early elementary students were also more attentive during their classes because they were eager to test their results using our simulator after finishing the electronic block based coding.

Below are some samples of student reflection assignments.

- "Coding is to issue commands to a computer."
- "Coding is a cat moving."
- "Scratch programming was fun. I also did it at home and had even more fun. I want to do it again."
- "Coding was really amazing. I assembled electronic blocks, which, in turn, drew a line."

In addition to our interest survey and interview, we conducted an assessment to test the understanding of programming concepts among our subjects.

The assessment of programming fundamentals tests if students understand the concepts such as sequences, loops, and conditionals. It consists of total nine questions (sequences: five questions, loops: two questions, and conditionals: two questions). Based on their types, we can also classify the questions into implementation (six questions) and error correction (three questions). Table 5 shows the details.

TABLE V: ASSESSMENT OF PROGRAMMING FUNDAMENTALS

Ques -tion	Contents	Concepts
1	Arranging blocks after observing the results	Sequences
2	Arranging blocks after observing the results	Sequences
3	Find errors after reviewing the code and outputs	Loops
4	Finding where a code snippet fits in an existing lines of code after reviewing the outputs	Sequences and error correction
5	Expecting what a sprite may do when touch a mouse pointer	Conditionals
6	Expecting what a sprite may do when touch a mouse pointer	Conditionals
7	Drawing a diagram representing an output after reviewing the code	Sequences
8	Drawing a diagram representing an output after reviewing the code	Loops
9	Regarding identifying errors and providing reasons after reviewing the code and its outputs	Sequences and error correction

Students had either short-answer or essay questions. Fig. 5. shows the number of students who answered the questions correctly.

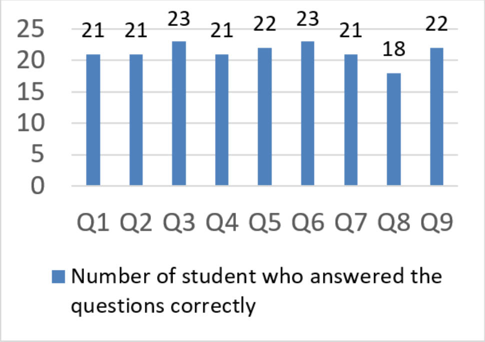


Fig. 5. The result of assessment.

Fourteen students(56%) completed all five questions including 'sequence' concept, and twenty one students(84%) completed more than four of the five questions. 17 students (68%) have completed all questions, including the 'loop' concept, and 22 (88%) have completed all questions, including the 'event' concept. Of the 25 students who participated in the assessment, 84% successfully completed seven or more questions out of nine provided, 61% percent of the students accomplished all nine questions.

V. CONCLUSION

In this research we described our tactile electronic block solution that makes Scratch more accessible to early elementary students. We implemented our solution in physical, tactile blocks students can touch and manipulate. Scratch electronic blocks are a tangible programming language tool. Children under nine, who have trouble in using a mouse and keyboard can still experience Scratch programming by assembling the electronic blocks as they do with LEGO building blocks. We conducted our assessment, and the results are quite encouraging. Student interest scores increased from 3.23 to 4.0. This also provides evidence that early elementary students can understand CT concepts such as sequences, loops, and conditionals.

In future, we plan to conduct classes for children under eight by using our Scratch electronic blocks. Our ultimate goal is to commercialize the prototype and popularize our product so that more early elementary students are exposed to a useful learning tool like Scratch.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Dr. Chun managed the project and designed the programming courses and co-wrote the paper. Ms. Jo supervised the programming class and analyzed the results. Dr. Ryoo analyzed the results and co-wrote the paper. All authors approved the final version.

Acknowledgment

We would like to thank Yeonju Go, a teacher at the Buyang elementary school for teaching the Scratch programming course and administering student surveys in her class. This research was supported by the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2019S1A5A2A03049067).

References

[1] N. B. Dohn, “Students' interest in Scratch coding in lower secondary mathematics,” *Br. J. Educ. Technol.*, vol. 51, no. 1, 2020, pp. 1-83.

[2] N. Smith, C. Sutcliffe, and L. Sandvik, “Code club: Bringing programming to UK primary schools through scratch,” *SIGCSE 2014*, pp. 517-522.

[3] H. L. Dankert, P. L. Davies, and W. J. Gavin, “Occupational therapy effects on visual-motor skills in preschool children,” *American Journal of Occupational Therapy*, vol. 57, no. 5, 2009, pp. 542–549.

[4] A. Donker and P. Reitsma, “Drag-and-drop errors in young children's use of the mouse,” *Interactive with Computers*, vol. 19, no. 2, 2007, pp. 257-266.

[5] CSTA, *Computer Science Framework*, 2016.

[6] P.-N. Chou, “Using scratchjr to foster young children’s computational thinking competence: A case study in a third-grade computer class,” *Journal of Educational Computing*, vol. 58, no. 3, 2020, pp. 570–595.

[7] V. Barr and C. Stephenson, “Bringing computational thinking to K12: What is involved and what is the role of the computer science education community?” *ACM Inroads*, vol. 2, no. 1, 2011, pp. 48–54.

[8] N. Villar, C. Morrison, D. Cletheroe, T. Regan, A. Thieme, and G. Saul, “Physical programming for blind and low vision children at scale,” *CHI Extended Abstracts*, 2019, pp. 1-4.

[9] Z. Y. Rong, N. F. Chan, T. Z. Chen, and K. N. Zhu, “CodeRhythm: Designing inclusive tangible programming blocks,” in *Proc. Conference on Designing Interactive Systems (Companion Volume)*, 2020, pp. 105-110.

[10] M. Alves, A. Sousa, and Â. Cardoso, “Web based robotic simulator for tactode tangible block programming system,” in *Proc. the 10th International Conference on Interaction Design and Children*, 2019, pp. 490-501.

[11] F. Reverter, “Toward Non-CPU activity in low-power MCU-based measurement systems,” *IEEE Trans. Instrum. Meas.*, vol. 69, no. 1, 2020, pp. 15-17.

[12] A. Butterfield and J. Szymanski, *A Dictionary of Electronic and Electrical Engineering(Oxford Quick Reference) Kindle Edition*, Oxford, 2018, p. 2570.

[13] S. C. Kong, M. M. Chiu, and M. Lai, “A study of primary school students' interest, collaboration attitude, and programming empowerment in computational thinking education,” *Computers & Education*, vol. 127, 2018, pp. 178-189.

Yunju Jo is Sin-mook elementary school teacher. From 2014 to 2020, she worked as an elementary teacher in Seoul Metropolitan Office of Education(SMOE). She is received the master`s degree in computer education from Seoul National University of Education(SNUE) in 2017. She is taking a Ph.D in computer education at SNUE. She is the master of SW education at the SMOE. Her current research interests include programming education, AI education, and online learning.

Seok-Ju Chun received the PhD degree in information and communication engineering from KAIST. From 1989 to 1995, he worked as a system engineer in Research and Development Center at Hyundai Heavy Industries(HHI), Korea. While at HHI, he developed ACONIS, an integrated ship automation system that monitors and controls the state of various machinery and equipment on board. From 1997 to 2004, he was an assistant professor in the Department of Internet Information at the Ansan University, Korea. Since 2004, he has been a Professor of Computer Education at Seoul National University of Education (SNUE), Korea. Dr. Chun is the PI of the "Development and Application of Coding Robot Kit for Silver Care in Aging Society" project supported by the NRF. His current research interests include programming education, AI education, OLAP, and data mining.

Jungwoo Ryoo is professor of information sciences and technology (IST) at Penn State Altoona. His research interests include innovative learning environments (ILEs) such as the use of virtual worlds in cybersecurity education. He led the NSF PI Forum on Virtual Environments and Game-Based Learning (GBL) in the Classroom. His NSF-funded project entitled “Immersive-Security Education Environment (I-SEE)” explored teaching and learning cyber security concepts in 3D virtual environments such as Second Life and Open Simulator. The findings from this research are available in the article, Security Education Using Second Life published in IEEE Security and Privacy. Dr. Ryoo is also the PI of an NSF-funded project whose goal is to build a big data analytics workforce in iSchools.