Tangible Programming Learning System (TPLS) for Programming Class with Visually Impaired and Sighted Students

Sean-Joo Kim*, Seung-Mee Lee, Seok-Ju Chun, and Jeong-Hyeon Seo

***Abstract*—In the inclusive classroom, the visually impaired and sighted student participate together in programming classes. However, familiar tools for the sighted students are difficult for the visually impaired students, and tools available to the visually impaired students are unfamiliar to the sighted students. These differences in tools make it difficult for these students to communicate and collaborate. To solve this problem, we developed a tangible programming system (TLPS). The system has an intuitive interface which can be used by both the visually impaired and the sighted people. To validate this system, 16 elementary school students participated in a usability evaluation and gave a positive evaluation of the system. In addition, the evaluation showed improvement in the participating students' disability awareness of the visually impaired and in programming.**

***Index Terms*—Visually impaired, programming learning system, accessibility, tangible, inclusive education**

I. Introduction

With the progression of the 4th industrial revolution, computer science, especially learning programming, has become important for everyone. This is also important for the visually impaired, and they also need programming education [1]. However, most computer education content and tools use a visual-dependent method, which is difficult for the visually impaired [2]. For example, they need to see and touch the screen, or take an input device using vision, for example, a keyboard or a mouse.

Because of this problem, a programming learning system for visually impaired people has been developed. A typical example is a tangible user interface (TUI) based system [3]. It allows users to program through physical objects, which makes learning much easier and more intuitive [4]. In particular, the system for the visually impaired does not use vision; rather it uses its own tangible method: recognizing the screen by sound or Braille [5–7].

However, depending on the educational environment, such a system can be an obstacle. This is because it requires additional learning on how to use it. This is not a problem in classes only for the visually impaired, but a problem in an inclusive classroom as well. Using unfamiliar tools makes it difficult for students to collaborate and communicate with each other. Likewise, it is also complicated for the teacher to teach using two disparate tools in the same class [8].

To solve this problem, we developed the Tangible Programming Learning System (TPLS). We made it usable to use for both the visually impaired and the sighted by using a tangible and intuitive interface. To test this system, 16 elementary school students participated and responded to a usability evaluation. Our hope is that visually impaired and sighted students share and collaborate in programming classes by taking our system.

Manuscript received August 19, 2022; revised September 5, 2022; accepted November 12, 2022.

Sean-Joo Kim, Seung-Mee Lee, and Seok-Ju Chun are with Seoul National University of Education, Republic of Korea (e-mail: km1817@naver.com).

Jeong-Hyeon Seo is with Sangtap Elementary School, Republic of Korea.

*Correspondence: km1817@naver.com (S.J.K.)

II. Related Work

As programming education becomes more important, various methods for programming education have been proposed. The most common method is to use a text-based or visual-based language [2]. However, since this method is visually dependent, it lowers the motivation of visually impaired students to learn or causes difficulties in learning [9]. Thus, in order to lower the programming barrier for the visually impaired, scholars are proposing innovative solutions. These can be classified into two categories: Using audio and Using tangible objects [3].

In case of audio, they turn the text-based or block-based languages into auditory. They are stand-alone tools or act as plug-in for existing tools. These programs use the new frame for screen readers, audio signals, or input/output replacement tools. For example, there is a programming editor that supports audio such as JavaSpeak, Aural Tree Navigator and CodeTalk [10–12]. It is an integrated development environment (IDE) that provides information of programs to users by voice. And there is a web-based method that is compatible with a screen reader like a Code Mirror Block [13]. Also, there are ways to increase accessibility to existing programs. To increase the accessibility of Google-developed Blocky, Stephanie *et al*. developed a new framework to make it compatible with screen readers and Caraco *et al*. studied how to access it through a touch screen and a screen reader method [14, 15].

In other cases, they use Braille or the tangible user interface (TUI). The TUI was developed by Hiroshi Ishii of the MIT Media Lab with the concept of a 'touching interface'. It refers to a method of expressing and manipulating information with actual physical objects, spaces, and shapes [16]. The tools developed using the TUI method are as follows. Blocks4All [5] is a tool that moves real robots by block coding using a new touch screen method. The user can recognize the block through sound by long-touching an element on the screen and checking the coding result by the actual robot movement. Moreover, Code Jumper, developed

doi: 10.18178/ijiet.2023.13.6.1896

by Microsoft's Project Torino [6], is a programming tool that makes music and sounds by connecting plastic command blocks with wires. Students use this tool to create musical elements and learn basic programming. JBricks [7] is a tangible programming tool based on logo blocks and a graphical user interface (GUI) from LEGO Mindstorms. It is easy to understand by assembling Braille blocks in order, and it can be freely combined and separated, and because the number of blocks is small, it is designed so that students can learn coding easily.

All of the tools were developed to suit the characteristics of the visually impaired. So, these tools are suitable for visually impaired people. However, some of these are unfamiliar to sighted people. There are times when the sighted need extra study and time to learn about the new touch screen or Braille. Therefore, we developed a programming education system that can be used intuitively by both visually impaired and sighted students without separate learning.

III. Tangible Programming Learning System

A. System Components

The components of this system are a 3D maze, tangible programming blocks, a programming block reader and a maze puzzle app as shown in Fig. 1.

The 3D maze is a 3D object converted from a maze puzzle problem in graphic form. There is a block corresponding to each stage of the maze puzzle. Each block is embossed with a question number, an arrow indicating the starting point and direction, and a star-shape destination. In addition, the path is also expressed in blanks (intaglio) so that the length and path of each maze can be recognized. The visually impaired can understand the problem by touching this block as shown in Fig. 2.

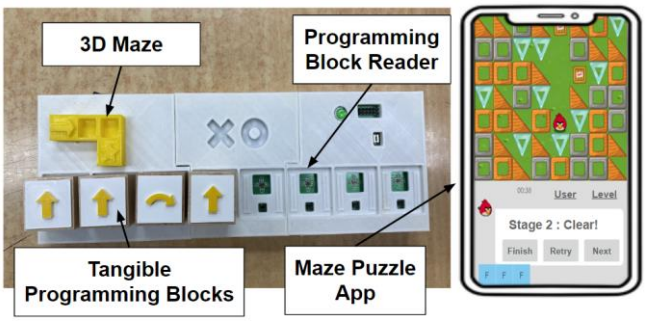


Fig. 1. Example of using components of TPLS.

Fig. 2. Using the 3D maze.

The tangible programming block is made of 3×3cm cubed wooden blocks. On the upper side of the block, arrows, rotation directions, numbers, and repeating symbols are embossed. On the underside, each block is painted with a unique color so that the programming block reader can identify each block through a color recognition method.

The tangible programming block is made up of essential blocks for sequence and iteration (Table I). Specifically, the types of programming blocks are move(↑), rotate(↷,↶), repeat(↻,■), and number(3,4,5) blocks. The repeat block consists of a start block and an end block, and is used as follows: If you want to go forward three steps, you can combine the move block and the number block and put them inside the repeat blocks as follows. (Example: ↻ 3 ↑ ■)

TABLE I: Tangible Programming Blocks

Type	Shape	Command
Moving Blocks	↑	Move forward
	↷	Turn right (90°)
	↶	Turn left (90°)
Iterating Blocks	↻	Begin of iteration
	■	End of iteration
Number Blocks	3, 4, 5	Set the number of iteration

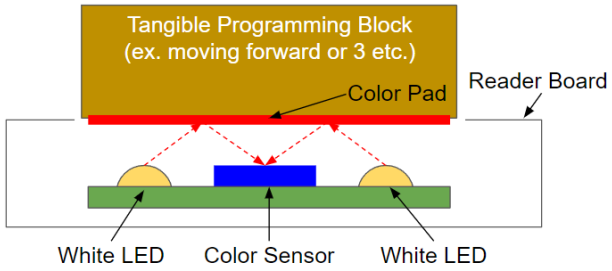


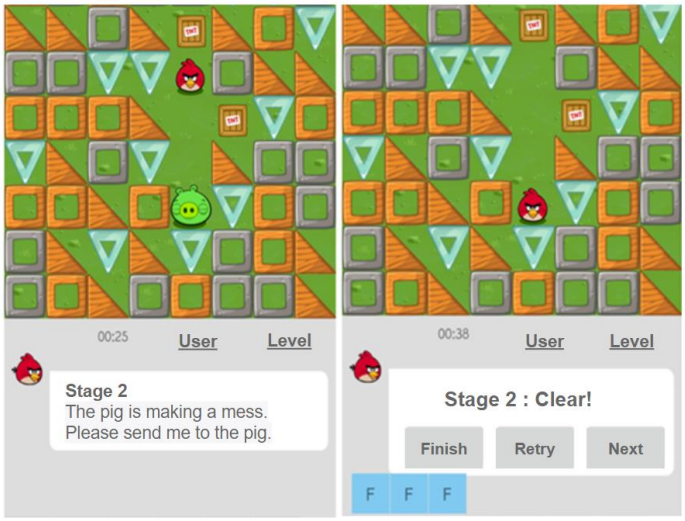
Fig. 3. Recognizing a block with a color sensor.

The programming block reader consists of a home for up to eight blocks, a send button, and a light-emitting diode (LED) to indicate correct and incorrect answers to a question. Under each groove, there are block detection sensors, an action button and a Bluetooth module. It also has a built-in color recognition sensor to recognize the tangible programming block. As shown in Fig. 3, a non-contact color recognition sensor was used.

The maze puzzle app was developed using Angry Birds puzzle maze game provided by code.org [17]. It consists of a total of 12 sequential and iterative programming problems. At the bottom of the screen, play times, stages, and missions are displayed (Fig. 4). You can start by designating a user when starting the app, and you can evaluate achievement by recording the user's play time for each level.

B. Programming Learning Process

The system developed in this study proceeds in three stages as shown in Fig. 5. First, in the <Step1: Exploration>, the user explores the maze puzzle problem. A visually impaired student uses a 3D maze and a sighted student can identify problems through the app screen. In <Step2: Problem solving>, users code the character to escape the maze by combining the tangible programming blocks and the programming block reader. Then, in <Step3: Execution>, the block programmed by the user is sent to the maze puzzle app and executed. Users can see the results through character movements or sounds in the maze puzzle app.



(a) First stage (b) Success stage

Fig. 4. Maze puzzle example (problem 2).

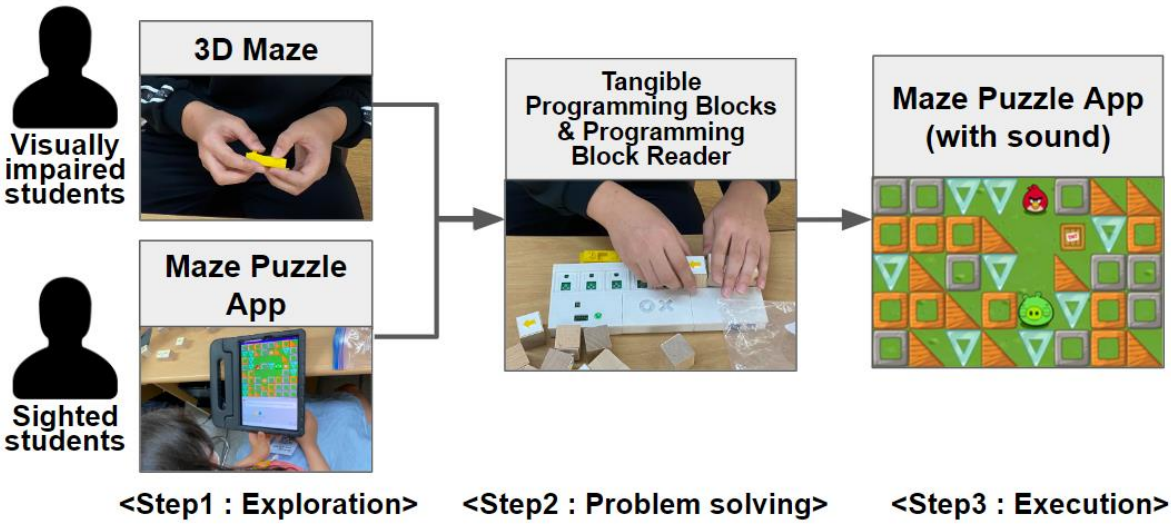


Fig. 5. Programming Learning Process using TPLS.

IV. Experiment and Result

There were two things we wanted to check in this experiment.

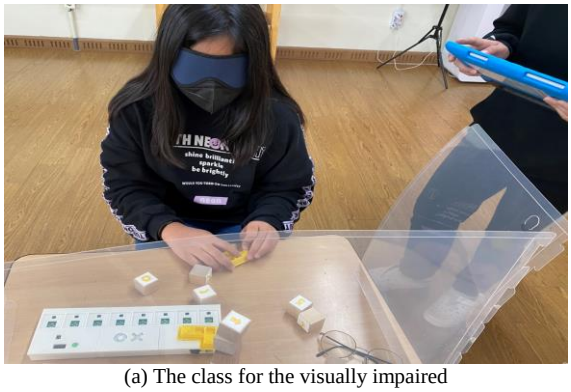
Research Question 1): Is this system useful for learning programming for both visually impaired and sighted users?

Research Question 2): Can both visually impaired and sighted users easily use this system without additional learning?

We experimented with the 4th class for two weeks in September and December 2021 at an elementary school in Seoul. The experiment involved 14 students aged 10 to 12 years old. To evaluate the system from the perspective of real users, we selected students with programming experience. We created two classes, a visually impaired class and a sighted class. Due to the COVID-19 pandemic, it was difficult to recruit the visually impaired students, so we had the sighted students wear eyepatches.

First, in the class for the visually impaired, two students paired up to conduct an experiment, one student wearing an eye patch and the other taking the role of an activity assistant. Schools for the visually impaired have one assistant teacher per student as shown in Fig. 6(a). The tutor student gave another student a 3D maze suitable for each step of the maze puzzle. The student solved the problems from steps 1 to 12 of the maze puzzle. On the other hand, the students of sighted class solved the maze puzzle after listening only to a brief explanation of the system as shown in Fig. 6(b).

To evaluate the system, the existing educational programming tool usability evaluation [18] was modified to the level of elementary school students. It consisted of 6 multiple-choice questions using a Likert scale (0–5 points).



(a) The class for the visually impaired

(b) The class for the sighted
Fig. 6. The classes with TPLS.

In the results of the survey, the overall satisfaction level of the classes for the visually impaired was 4.58 and the sighted class was 4.43. Since the satisfaction of the two groups was located between 4 (yes) and 5 (very much), the overall satisfaction with this system can be judged as positive. Table II shows the specific results.

TABLE II: USABILITY EVALUATION RESULTS FOR TPLS

	Questions	the visually impaired		the sighted	
		M	SD	M	SD
1	Usefulness	4.75	0.50	4.42	1.14
2	Diversity	4.00	1.41	4.00	1.14
3	Step-by-step learning	4.75	0.5	4.38	0.27
4	Accuracy	4.75	0.50	4.38	0.27
5	Safety	4.75	0.50	4.63	0.27
6	Ease	4.5	1.41	4.75	0.50
	Total	4.58	0.80	4.43	0.60

On the usefulness of this system, both groups showed high satisfaction with 4.75 and 4.42, and the score for step-by-step learning was high at 4.75 and 4.38. Moreover, in accuracy and stability, the visually impaired group gave a high score of 4.75, and the sighted group gave 4.38 and 4.63. On the other hand, both groups showed low satisfaction with the diversity of the professor at 4.0, because this system is not free programming but a game for learning programming elements. However, there were no items below 4 points, which confirmed that this system is suitable as a programming learning system for both the visually impaired and the sighted.

In addition, as for the ease of operation, positive results were shown at 4.5 and 4.75, indicating that both the visually impaired students and the sighted students were able to use the system without any difficulties in learning or operation.

Meanwhile, in this experiment, non-visually impaired students evaluated the tools from the perspective of the visually impaired, so to more clearly analyze the effects of this experiment, we conducted additional interviews. The results are shown in Table III.

Before the experiment, most of the students were of the opinion that programming with the blind was “never thought about it, and it would be very difficult or impossible”.

TABLE III: INTERVIEWS WITH VISUALLY IMPAIRED CLASS STUDENTS

(Before the experiment)

What do you think about programming for the visually impaired?

- Student 1 : I never thought about it.
- Student 2 : I thought they couldn’t.
- Student 3 : It will be very difficult or in Braille.
- Student 4 : They will not be able to do it on their own and will need help.
- Student 5 : Coding will take a long time because you can’t see it.
- Student 6 : I think they use the brain wave system.

(After experiment)

How do you feel after receiving programming education for the visually impaired?

- Student 1 : Because we did well when we put on the eyepatch, they will do well programming, too.
- Student 2 : It is very easy and simple to use, so it is very suitable for the visually impaired.
- Student 3 : I came to think that even the visually impaired can program.
- Student 4 : I thought that people with disabilities could do the same.
- Student 5 : The visually impaired also want to become a programming-related profession or do programming.
- Student 6 : In order to give such people opportunities, the system or tool that allows visually impaired people is essential.

However, after the experiment, the students responded that, based on their experience of participating in the experiment, “visually impaired people will be able to learn programming sufficiently by using this system”. In addition, they said, “Even the visually impaired can program without any difference from the sighted, and such rights and opportunities should be given.” Furthermore, there were students who realized the need for systems and tools for this.

Overall, we found that this system can be used easily by both the visually impaired and the sighted. Furthermore, the experience of this system improved the students’ disability awareness of the visually impaired and programming.

V. CONCLUSION

In this study, we developed a programming learning system that can be used by both the visually impaired and the sighted. It uses the tangible user interface (TUI) method and can be recognized visually and intuitively. And this system is consisted of a 3D maze, tangible programming blocks, a programming block reader and a maze puzzle app. Learners can learn the basics of programming step by step through this maze puzzle.

To evaluate this system, we conducted an experiment targeting 6th graders of elementary school. When this was tested, it received high scores in the usability evaluation and was evaluated as simple and easy for both the visually impaired and the sighted. In addition, the students who participated in the class improved their perception of the visually impaired and programming.

In the future, we plan to conduct classes for diverse students with visual impairments. We expect that this study will allow both visually impaired and sighted people to share and collaborate in programming classes.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Dr. Chun managed the project and designed the programming courses. Ms. Kim taught the programming courses and administered students survey. Ms. Lee and Dr. Seo supervised the programming class and analyzed the results. All authors co-wrote the paper and approved the final version.

Acknowledgement

This research was supported by the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2021R1F1A1046712).

References

[1] M. W. Jeannette, "Computational thinking," *Communication of the ACM*, vol. 49, no. 3, pp. 33–35, March 2006.

[2] A. Hadwen-Bennett, S. Sentence, and C. Morrison, "Making programming accessible to learners with visual impairments: A literature review," *International Journal of Computer Science Education in Schools*, vol. 2, no. 2, pp. 3–13, May 2018.

[3] M. Aboubakar, O. Obianuju, and L. Stephanie, "Addressing accessibility barriers in programming for people with visual impairments: a literature review," *ACM Transactions on Accessible Computing*, vol. 15, no. 1, pp. 1–26, March 2022.

[4] B. David, G. Jeff, K. Caitlin, S. Josh, and T. Franklyn, "Learnable programming: Blocks and beyond," *Communication of the ACM*, vol. 60, no. 6, pp. 72–80, June 2017.

[5] L. R. Milne and R. E. Ladner, "Blocks4all: Making blocks-based programming languages accessible for children with visual impairments," in *Proc. the 50th ACM Technical Symposium on Computer Science Education*, p. 1290, February 2019.

[6] C. Morrison, N. Villar, A. Thieme, Z. Ashktorab, E. Taysom, O. Salandin, D. Cletheroe, G. Saul, A. F. Blackwell, D. Edge, M. Grayson, and H. Zhang, "Torino: A tangible programming language inclusive of children with visual disabilities," *Human–Computer Interaction*, vol. 35, no. 3, pp. 191–239, Oct. 2018.

[7] S. Ludi, M. Abadi, Y. Fujiki, P. Sankaran, and S. Herzberg, "JBrick: Accessible lego mindstorm programming tool for users who are visually impaired," in *Proc. the 12th International ACM SIGACCESS Conference on Computers and Accessibility*, pp. 271–272, Oct. 2010.

[8] J. Pivik, J. McComas, and M. Laflamme, "Barriers and facilitators to inclusive education," *Exceptional Children*, vol. 69, no. 1, pp. 97–107, 2002.

[9] M. G. Funk, J. M. Cascalho, A. I. Santos, and A. B. Mendes, "Educational robotics and tangible devices for promoting computational thinking," *Frontiers in Robotics and AI*, vol. 8, Article 713416, Nov. 2021.

[10] A. C. Smith, J. M. Francioni, and S. D. Matzek, "A java programming tool for students with visual disabilities," *Assets '00: Proceedings of the 4th International ACM Conference on Assistive technologies*, pp. 142–148, Nov. 2000.

[11] A. C. Smith, J. S. Cook, J. M. Francioni, A. Hossain, M. Anwar, and M. F. Rahman, "Nonvisual tool for navigating hierarchical structures," *ACM SIGACCESS Accessibility and Computing*, issue 77–78, pp. 133–139, 2003.

[12] V. Potluri, P. Vaithilingam, S. Iyengar, Y. Vidya, M. Swaminathan, and G. Srinivasa, "CodeTalk: Improving programming environment accessibility for visually impaired developers," in *Proc. the 2018 CHI Conference on Human Factors in Computing System*, no. 618, pp. 1–11, 2018.

[13] E. Schanzer, S. Bahram, and S. Krishnamurthi, "Accessible AST-based programming for visually-impaired programmers," in *Proc. the 50th ACM Technical Symposium on Computer Science Education*, pp. 773–779, Feb. 2019.

[14] L. Stephanie and S. Mary, "Design considerations to increase block-based language accessibility for blind programmers via blockly," *Journal of Visual Languages and Sentient Systems*, vol. 3, pp. 119–124, July 2017.

[15] B. Caraco, S. Deibel, Y. Ma, and L. R. Milne, "Making the blockly library accessible via touchscreen," in *Proc. the 21st International ACM SIGACCESS Conference on Computers and Accessibility*, pp. 648–650, Oct. 2019.

[16] B. Ullmer and H. Ishii, "Emerging frameworks for tangible user interfaces," *IBM Systems Journal*, vol. 39, no. 3. 4, pp. 915–931, July 2000.

[17] CODE.ORG. (Nov. 15, 2021). [Online]. Available: https://code.org

[18] J. Á. Velázquez-Iturbide, A. Pérez-Carrasco, and O. Debdi, "Experiences in usability evaluation of educational programming tools," *STEM Education: Concepts, Methodologies, Tools, and Applications*, pp. 461–480, 2015.