

# Evaluation of a Classroom Support System for Programming Education Using Tangible Materials

Koji Oda<sup>1</sup>, Toshiyasu Kato<sup>2</sup>, Yasushi Kambayashi<sup>3</sup>

<sup>1</sup>Department of Information and Telecommunication, Saitama Prefectural Kawaguchi Technical High School, Kawaguchi, 333-0846, Japan

<sup>2</sup>Department of Information and Media Engineering, Nippon Institute of Technology, Minamisaitama, 345-0826, Japan

<sup>3</sup>Department of Informatics and Data Science, Sanyo-Onoda City University, Sanyo-Onoda, 756-0884, Japan

## ARTICLE INFO

Article history:

Received: 14 September, 2024

Revised: 09 October, 2024

Accepted: 08 October, 2024

Online: 18 October, 2024

Keywords:

Tangible materials

Programming education

Classroom support systems

Face-to-face instruction

## ABSTRACT

*In recent years, the utilization of tangible educational materials has attracted attention on educational settings. They provide hands-on learning experiences for beginners. This trend is especially notable in the field of programming education. Such educational materials are employed in many institutions worldwide. They liberate learners of programming from programming languages that are confined in a small computer screen. On the other hand, in the school setting, classroom time is limited. When instructing more than thirty students, it is hard for instructors to provide adequate guidance for everyone. To address this problem, we have developed a classroom support system for programming education that complements the use of tangible educational materials. With this system, instructors can monitor the real-time progress of each student during the class and analyze which parts of the program many students find challenging. Based on these analytical results, instructors can provide appropriate instructions for individual students and effectively conduct the class. This system is suitable for programming education in high schools. It quantifies each student's ability of programming and track the progress of each student. We administered a questionnaire to both the students and the instructor. The results of the questionnaire show our system is well received by both students and the instructor. Even though our system demonstrates some usefulness for programming beginners, we are aware that our system has some serious limitations such as our rigid model answers.*

## 1. Introduction

This paper is an extension of work originally presented in 2024 Twelfth International Conference on Information and Education Technology (ICIET 2024) [1]. The work presented the basic idea of system and the results of the preliminary experiments that indicated its usefulness. In this paper, we have extended the paper to explain our system in details and to demonstrate its effectiveness through showing results of larger scale experiments. For programming beginners, numerous GUI programming systems have been proposed. However, the computer screen and the display resolution restrict the students' recognizability of program elements. This problem makes the programming activities difficult

especially with lower resolution displays. To address this issue, we developed tangible educational materials named "Jigsaw Coder" for programming education [2]. In the following, we will refer to this as JC. JC consists of multiple cards. Each card has QR code printed on it, and students can construct programs by rearranging them. This enables programming on a desk or even on the floor, which provides much larger space. The user can take a photo to read the complete program by their smartphones and also execute the program on their smartphone. However, such tangible educational materials were designed for self-taught of individual learners. It is challenging for class room use; it is hard for instructors to grasp the progresses of all students when used in a class of more than a few, e.g. thirty, students. The objective of this study is to design and to implement a system that provides instructors information of real-time progresses of the students so

\*Corresponding Author: Koji Oda, Saitama Prefectural Kawaguchi Technical High School. [mooda194lun@yahoo.co.jp](mailto:mooda194lun@yahoo.co.jp)

that he or she can analyse information of students' programming in classes using JC. The system helps instructors to practice much effective use of instruction time.

The authors conducted a preliminary evaluation of JC as prior research [1]. As a result of performing a functional check assuming an actual class, there were no issues with the system's operation with around ten users, and it was possible to conduct a trial evaluation simulating an actual class. This paper demonstrates the effectiveness of JC through an evaluation experiment conducted in actual high school classes.

## 2. Research Methodology

We have developed a tangible programming system that utilizes JC and Micro:bit for educational purpose. This system allows students to engage in tangible programming, while instructor can monitor their progresses in real-time and perform analysis over their achievements. Subsequently, we conducted classes as part of the evaluation experiments and administrated questionnaires for both instructor and students to assess the effectiveness of the system. In the previous papers, we reported our development and evaluation of the tangible educational materials [3, 4]. The materials involve rearranging multiple cards to program. Then the user makes the system read the QR codes printed on them using a smartphone to execute the program. We call this card-type tangible educational system as JC. Figure 1 shows the flow of the programming process. In the original JC, we used a smartphone; in this study, we decided to utilize Chromebooks, because they are easy to use and widely adopted in many Japanese schools.

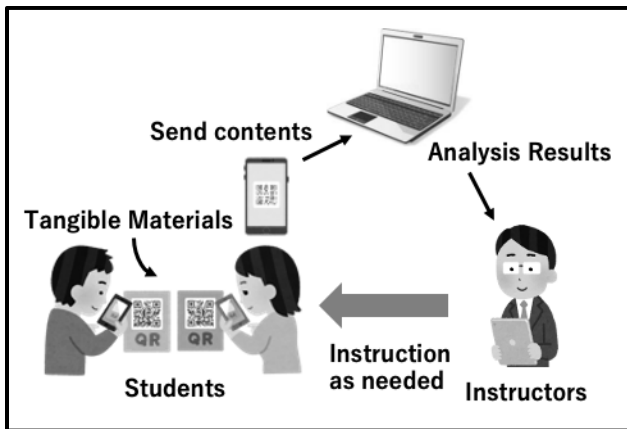


Figure 1: Instruction utilizing tangible education materials

### 2.1. JC

A client PC (Chromebook) creates a program from QR cards and writes it to the Micro:bit. Simultaneously, the program code is transferred to the server. The server then analyzes the received program code. The analysis flow is as follows.

The server saves the program code as a file and compares it with the corresponding model answer. In the comparison process, it calculates the matching rate with the model answer and identifies the positions of incorrect sections. The answer data for each student--such as student name, first answer time, most recent answer time, final answer time, number of responses, matching rate with the model answer, line numbers and positions of mistakes,

and program level--is stored in the database. Subsequently, a web page reads the database and displays the answer information for each student. At this point, based on the answer information, students are classified into three categories: Unanswered, Progress, and Completed. This allows the instructor to easily track each students' progress at a glance. Additionally, a page is generated that allows the instructor to review each students' answer. On this page, it is easy to identify missing, extra, or incorrect parts of the answer. Based on this information, the instructor can provide specific feedback to the students.

### 2.2. Micro:bit

Micro:bit is a microcontroller designed by the British Broadcasting Corporation (BBC) for programming education. It can display characters and shapes on LEDs and produce sound through a speaker. It also features sensors such as an accelerometer, magnetometer, microphone, temperature sensor, and light sensor, which allow it to recognize vibrations and changes in its environment. Additionally, Micro:bit includes wireless communication capabilities, enabling it to communicate with other Micro:bit. Programming can be done via a browser or app, and programs can be transferred to the Micro:bit for execution. Figure 2 shows a Micro:bit.

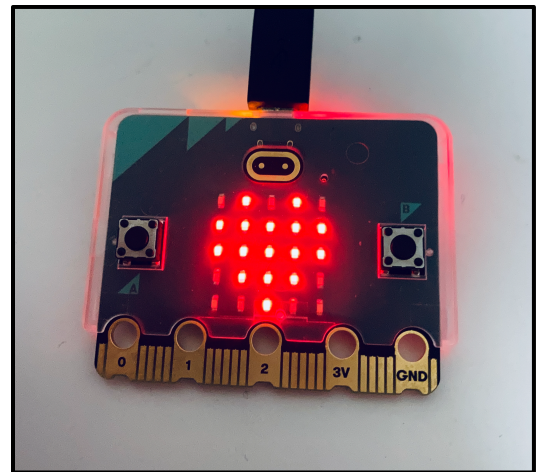


Figure 2: Micro:bit

## 3. Design And Implementation

We developed our system using Python. The reason for choosing Python is its high readability due to a vast array of libraries. This fact let us build shorter programs. In addition, Python is an interpreted language, enabling immediate execution without compilation, making it suitable for creating prototypes. To run the proposed system, some preparations are needed. The preparation before the class includes:

- 1) Creating tasks for students (assigning unique task numbers).
- 2) Creating and placing example answer programs and level configuration files.
- 3) Inputting students' information.

Carrying out a programming class includes:

- 1) Starting the server and server program.

- 2) Connecting Micro:bit to student's Chromebook.
- 3) Starting the client program.

### 3.1. Improvement of Jigsaw Coder

In this project, we added three more elements to enable more intuitive rearrangements. The first element is emphasizing the task number. To distinguish which task the student is working on, he or she initially needs to make the system read the QR code for the task number card in JC. Then, the background color of the task number card was changed, and highlighted the numbers by surrounding them with star symbols. The second element is the use of symbols “▷” and “◁”. These symbols represent the role of “{” and “}” in the conventional programming languages such as C and Java. They are used to denote looping constructs like “Repeat ▷” and “◁ End here,” aiding in the intuitive understanding of grouping. The third element is “→” and “←”, representing arranging cards side by side. These symbols are utilized when specifying conditional statements, such as “If Condition →” and “← Press A Button ▷”. These symbols help learners intuitively grasp the utilization and representation of conditions. Figure 3 shows the cards used by the students.

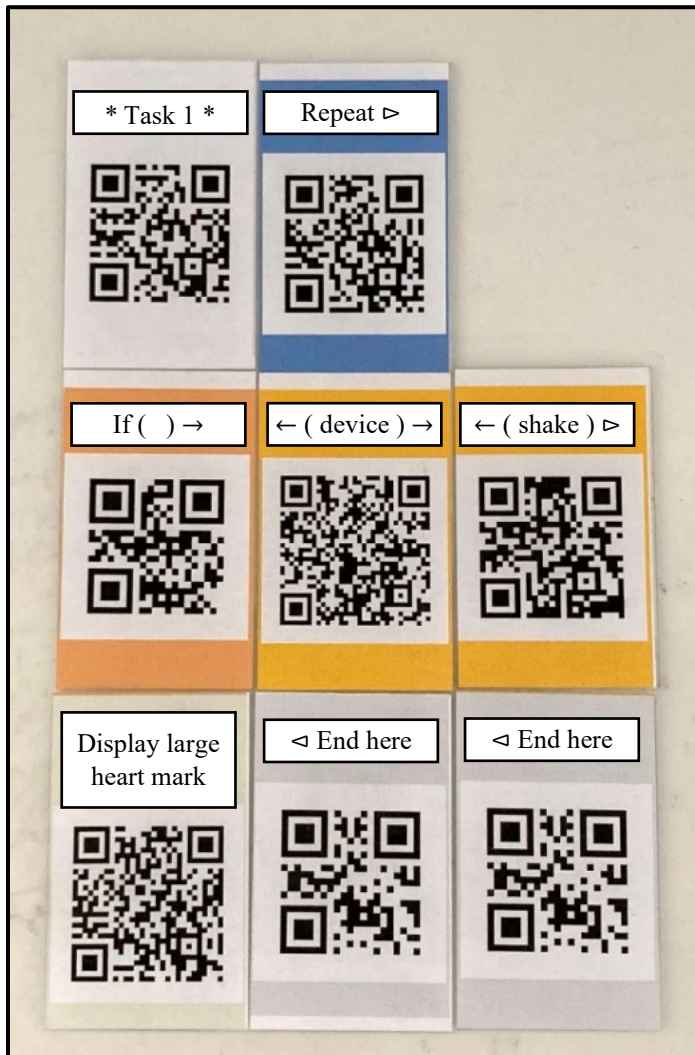


Figure 3: QR cards used in JC

### 3.2. Operation of the Students' Side (Client Program)

Figure 4 shows the flow of operations for the client program.

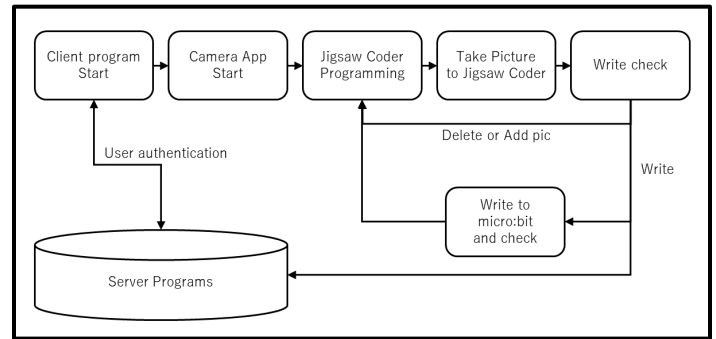


Figure 4: Flow of client program

Upon starting the client program, student authentication is initiated. The system prompts the student to input the grade, class, and the student number. Upon pressing the confirm button, the connection with the server program is established, and the students' name is displayed. Figure 5 shows the user authentication window.

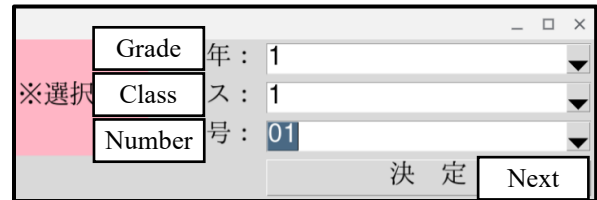


Figure 5: User authentication window

After completing student authentication, students begin programming. Once the students complete their arrangements of the cards, they photograph the cards using the camera application. The client program reads the captured photo, analyzes the QR codes in order, and generates the corresponding program. The captured photos are deleted to save memory space as they are no longer needed. Students can review the generated program in a window and then write it to the Micro:bit after confirmation. Figure 6 shows the confirmation window.



Figure 6: Writing confirmation window

If “Read additional” is selected, the read program is temporarily saved, and the student can capture another photo as the continuation of the program using the camera application. If “No (initialize content)” is selected, the read program is deleted, and the students can take a new photo again from the beginning. If “Yes” is selected, the system initiates the writing process to the



Micro:bit connected to the Chromebook. At this point, the student sends the program they wrote to the server program. The student checks their Micro:bit to ensure that the program is running correctly. If errors are found, the student rearranges the cards and takes another photo again. Figure 7 displays a photo of the system used by students.

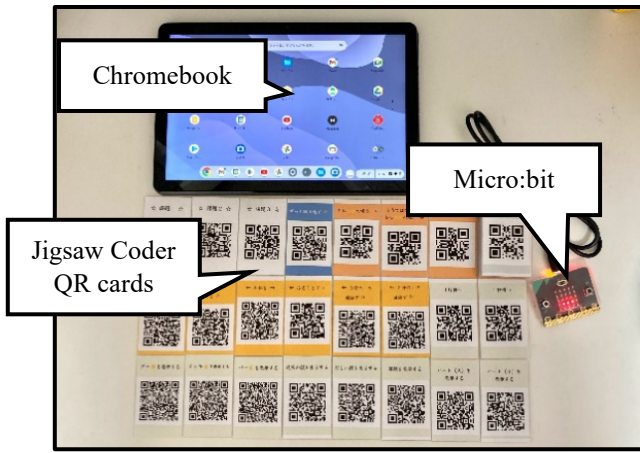


Figure 7: Overview of the student side system of JC

### 3.3. Operation of the Instructor's Side (Server Program)

Within the server, a database is set to manage a list of students and multiple tasks for them. The database contains a table for the student list, where their information is pre-stored for student authentication purposes. On the other hand, the task table maintains student progress, including the date and time the student first answered, date and time of subsequent attempts, date and time of correct answers, number of attempts, position of incorrect parts of their programs compared to the example answer program and level configuration file, match rate, and the program level calculated from the level configuration file.

### 3.4. The Web Page for Analysis

The instructor reviews the information in the database on a web page using a browser. This web page accesses the database using PHP and presents the information in an easy-to-review format for the instructor. Figure 8 shows the top page, where the number of programs in progress and completed answer programs for each task are summarized in a table. From the student list page, the instructor can edit or delete student information. It is also possible to import student data from an Excel file.

Each task page consists of two segments. Figures 9 and Figure 10 show the pages for a task. At first, using the first segment, instructor analyzes the parts of the program where students frequently make mistakes. The segment displays an example answer program, highlighting the background color of the areas where many students make mistakes. The background color changes from blue to yellow to red as the number of students who make mistakes increases in that particular section. The second segment is the table summarizing the progress of each student. It is divided into three tables for not answered, in progress, and completed of the programs, compiling details such as date and time of answer, number of attempts, and positions of errors. Using this table, the instructor can grasp the progress of the entire class. Additionally, by selecting a students' ID in this table, the instructor

can see a page that compares the students' program and the example answer program for that specific student. Figure 11 shows the comparison page.

Project T			
Task 1, 2, 3			
課題名	【課題 1】	【課題 2】	【課題 3】
Total	14	14	9
Progress	0	3	2
Completed	14	11	7
ユーザ設定 Student list			

Figure 8: Analysis table on the homepage

```

◆解答分析◆ Analysis
from microbit import *
import random
#que_1
while True:
    display.show(Image.HEART)
    sleep(1000)
    display.show(Image.HEART_SMALL)
    sleep(1000)
    
```

Yellow (points to the first sleep function)

Blue (points to the second sleep function)

Figure 9: Segment for analyzing answer

Unanswered	
No.	Name
1113	Chloe Graham
1115	Oliver MacLeod

Progress										
No.	Name	Start	Latest	Finish	Count	Match	Error Line	Error Pos.	Level	Other
1106	Megan Mackenzie	10:17	10:17		1	96%	8行目	1文字目	2	
1108	Amanda Paige	10:18	10:18		1	96%	8行目	1文字目	2	
1111	Sophie Ross	10:21	10:21		1	96%	8行目	1文字目	2	
1107	Elizabeth Harris	10:16	10:16				8行目	1文字目	1	
1102	Lisa Arnold	10:17	10:17				8行目	2文字目	2	
1103	Keith Thomson	10:17	10:17				8行目	2文字目	2	
1109	Paul Hunter	10:19	10:19		1	90%	6行目	2文字目	2	
1110	William Peake	10:19	10:19		1	90%	6行目	2文字目	2	
1112	Joan Bond	10:21	10:21				6行目	2文字目	2	
1114	Kylie Bower	10:22	10:22				6行目	2文字目	2	

The error is in line 8.

The error is in the second character.

Completed										
No.	Name	Start	Latest	Finish	Count	Match	Error Line	Error Pos.	Level	Other
1101	Penelope Morrison	10:09	10:09	10:09	1	100%				
1104	Felicity Oliver	10:16	10:16	10:16	1	100%				
1105	Sophie Fraser	10:16	10:16	10:16	1	100%				
9999	管理者	15:42	15:42	15:42	1	100%				

Figure 10: Segment for display answers list



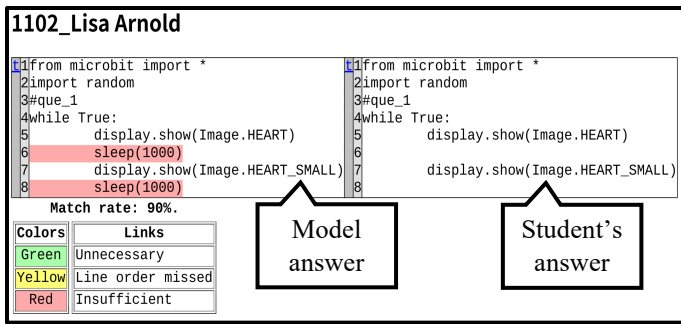


Figure 11: Page for comparing model answer with students' answer

#### 4. Evaluation Experiment

In order to demonstrate the effectiveness of our system, we conducted evaluation experiments on the system. The objectives of the experiments are as follows.

- To determine whether the system can function properly for a large number of students during actual class time.
- If any delays occur, to measure the duration of these delays.
- From the students' perspective, we evaluated "Overall system feedback," "Feedback on sequential processing, repetition, and branching in programming," and asked "Whether programming beginners can develop an interest in programming."
- From the instructor's perspective, we evaluated "Overall system feedback," "Convenience of monitoring student progress," and asked "Areas of potential improvement in the system."

To confirm the above objectives, we let a high school instructor conduct an actual programming class. Afterwards, we administered questionnaires to both the students and the instructor. The students are nineteen first-year students from Gunma Prefectural Annaka General Academic High School. The students had no prior programming experience. The tasks prepared for this evaluation experiment were as follows:

- 1) Display a large heart mark and a small heart alternately for 1 second each.
- 2) Pressing the "A" button when the device displays a smiling face, pressing the "B" button when it displays a sad face, and not pressing any button when it displays a neutral face.
- 3) Shaking the device displays either rock, paper, or scissors for a rock-paper-scissors game.

The objective of task 1 is to facilitate learning of sequential processing and repetition. Task 2 aims to utilize button inputs and learn about branching. Task 3 is an optional task. The goal of this task is to utilize shaking the device as an input and understand the multi-level branching in the context of learning. All tasks involve elements of repetition.

#### 4.1. Experimental Results

We found the system is stable. The instructor felt no perceivable delays. However, on the client side, there were instances where the program stops due to students' input errors.

#### 4.2. Results of the Student Survey

The followings are the questions for the student survey:

- 1) Did using this material help you grasp the basic structure of a program?
- 2) Did experiencing this material increase your interest in programming?
- 3) Please indicate your perceived level of understanding of sequential processing.
- 4) Please indicate your perceived level of understanding of iterative (repetitive) processing.
- 5) Please indicate your perceived level of understanding of conditional processing.
- 6) Please write what you felt on the materials and lessons used in this session.

The questions and answers from the student survey correspond to Figures 12 through 16.

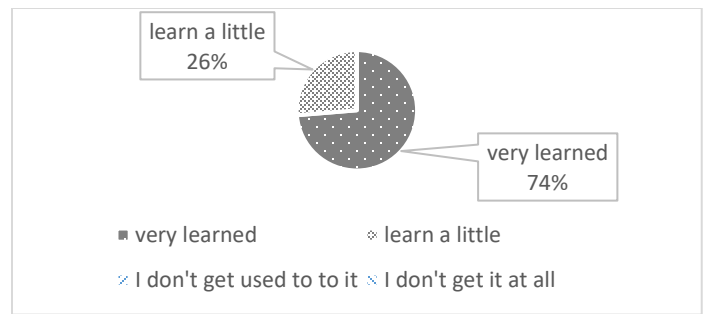


Figure 12: Did using this material help you grasp the basic structure of programming?

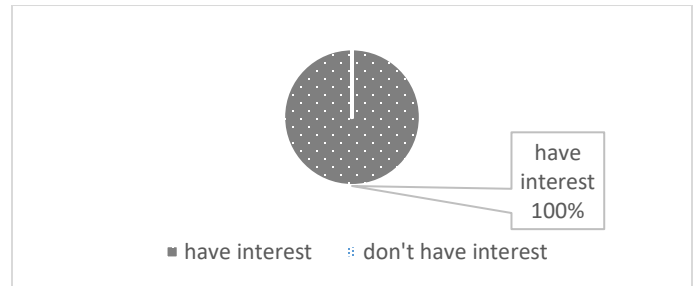


Figure 13: Did experiencing this material increase your interest in programming?

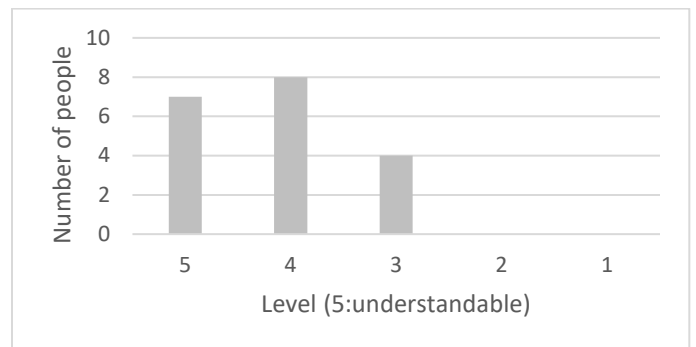


Figure 14: Please indicate your perceived level of understanding of sequential processing.

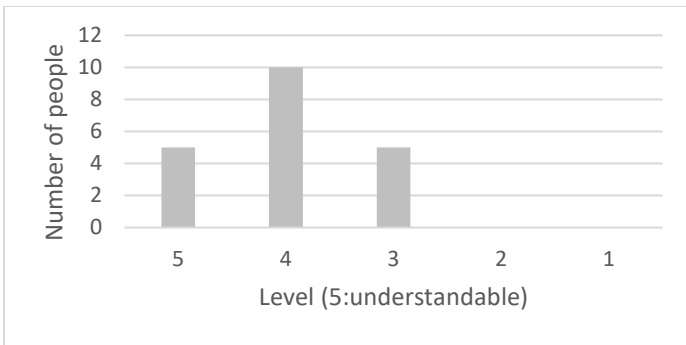


Figure 15: Please indicate your perceived level of understanding of iterative (repetitive) processing.

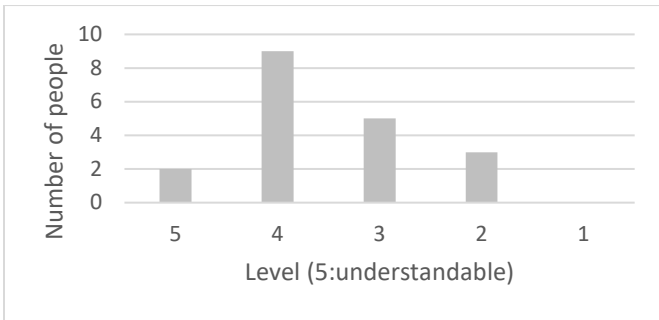


Figure 16: Please indicate your perceived level of understanding of conditional processing.

As an answer to question 6 (Please write what you felt on the materials used in this class.), the following feedbacks were provided:

- The steps were easy to remember.
- The cards were heavy.
- Having the materials allowed for better communication between students and instructors.
- I was able to think and work on my own.
- I couldn't solve the conditional processing problems.

#### 4.3. Results of the Instructor Survey

After class, we conducted a survey for the instructor. The questions and answers are as follows. Questions A through C use a 4-point scale, where 1 indicates the lowest rating and 4 indicates the highest rating, respectively. Table 1 shows the results.

- Did you comprehend the overall class situations based on the presented analysis results?
- Did you discover specific problems based on the presented analysis results?
- Did the classification based on learning levels from the presented analysis results assist for the instruction?

Table 1: Results of the survey for instructor

Question	Answer
A	2
B	3
C	1

D) Please tell us what you like about this system.

- The tangible materials, involving the combination of physical cards, are promising as an introductory tool for those new to programming.
- Taking photos of the program cards is easy and accurate enough.
- We can monitor students' progresses without moving around the classroom.
- We can focus on the students with many errors.
- The three tasks within a two-hour class is appropriate.
- Instead of using the system for real-time monitoring during class, it might be beneficial as a self-learning tool. Results, including errors, could inform instruction for future classes.

E) Please tell us any dissatisfactions or points for improvement of this system.

- It is difficult to take pictures because of the wired Micro:bit connection.
- There were many connection errors with the Micro:bit. It needs to be improved. It was hard to tell whether it is a connection error or a programming error. It would be good to have an indication lamp or beep sound for that.
- I is unable to identify which part of the program students are struggling.
- When instructors inspect students' programs, they see the corresponding Python code instead of JC cards. It is stressful for instructors without sufficient programming skills.

## 5. Discussion

This section analyzes and discusses the results of the evaluation experiments.

### 5.1. Discussion Based on the Student Survey

Survey results indicate positive feedback on the materials. Question 2 reveals that our system successfully developed intellectual inquisitiveness for programming in all the students. Since all the students were beginners, the system effectively achieved its goal of generating motivation for programming. For question 3 on sequential processing, there were many positive responses. Students understood the order of operations by rearranging the cards. This suggests that the card arrangement helped clarify sequential processing. Question 4 on iterative processing also received positive feedbacks. In contrast, question 5 on conditional processing had a lower average rating of 3.53. This lower rating may be due to the task's difficulty. Task 3, designed to teach conditional processing, required two conditionals, which might be challenging. Starting with simpler tasks could improve understanding of conditional processing. Additionally, students might have struggled with the visual and intuitive differences between "if" and "else if," as well as between "→" and "▷." We need to reconsider the design of JC to make these concepts more intuitive and easier to grasp. We plan to have different notations in the next version.

### 5.2. Discussion Based on the Instructor Survey

We can obtain several insights from the instructor survey. Questions A and C received negative responses. They indicate problems with the current system. Although instructors could track students' progress without moving around the classroom, they struggled to identify overall student difficulties. Positive responses to question B show the system is effective in identifying issues of individual students. However, question E responses indicate that how the instructor feel the system depends on their background knowledge of Python. The system requires instructors to read Python code on their screens, which may be problematic if their programming skills are insufficient. We may need to reconsider our assumption that the instructor should have sufficient programming skill in Python, and how to show students' progresses to the instructors. We plan to forge a novel means to display students' progresses so that it enhances the system's accessibility and effectiveness for users with varying levels of programming expertise.

### 5.3. Discussion of the Overall System

The system has not faced performance issues like delays so far. However, future experiments with over thirty students may present challenges. Unforeseen issues such as delays could arise depending on the server's capacity. Currently, a LAMP environment on a Chromebook is used for testing, but a dedicated server may be needed for practical use. Processing programs also needs adjustment to accommodate more number of users. Ensuring the system remains functional even when users cause serious runtime errors is crucial. For instance, adding confirmation dialogs to prevent accidental stops of the client program could reduce such opportunities. The card recognition issues, such as when only nine out of ten cards are recognized due to environmental factors, suggest the need for improving such as providing a new confirmation window.

We are implementing such a confirmation window. Figure 17 shows the new confirmation window that replaces the one shown in Figure 6. Since the message displayed in this window is written in Japanese, we show the corresponding English translation in Figure 18. Displaying the text on the cards before transferring to the Micro:bit could ensure correct card recognition. This approach helps students review their work and strengthen their understanding of both tangible and text-based programming.

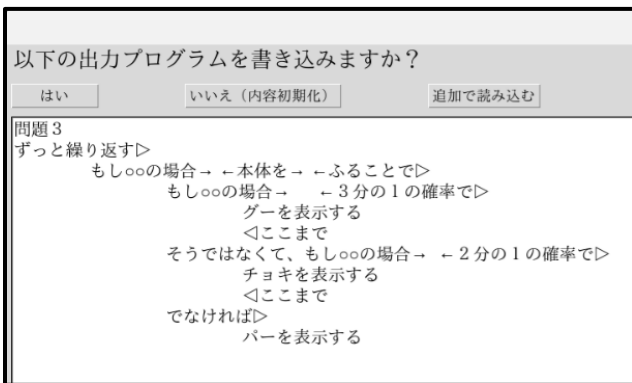


Figure 17: Writing confirmation window in Japanese

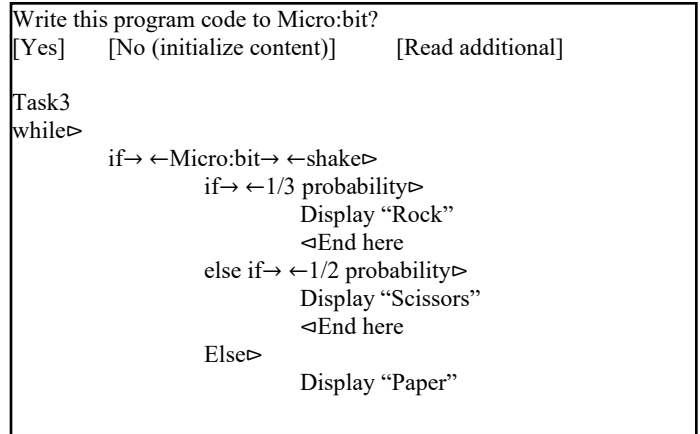


Figure 18: Writing confirmation window in English

The current limitation of this system is that only one model answer can be set for each task. Since current problem set includes only simple problems, one model answer for each problem is sufficient. We plan to incorporate a parser in our system, and to utilize AI to interpret responses more flexibly. This could identify not only syntax errors, but also semantic errors and runtime errors and provide tailored feedback for each program. Figure 19 shows an example of generated advice. While this example uses a GUI-based ChatGPT, we plan to incorporate the Python API for faster processing.

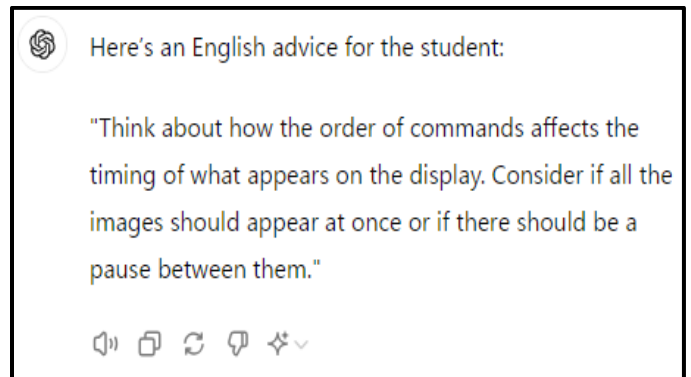


Figure 19: An example of advice generation using ChatGPT



Table 2: Comparison with other tangible educational materials

Name	Tangible	Analysis Multi-Student	Guidance for Struggling Students	Analysis on Class	Analysis after Class	Generating Individual Feedback
Jigsaw Coder	+	+	+	+	+	-
T-Maze [5]	+	-	-	-	-	-
Plugramming [6]	+	-	-	-	-	-
Strawbies [7]	+	-	-	-	-	-
PaPL [8]	+	+	-	-	+	-
Kamichi's System [9]	-	+	-	+	+	+
Kato's System [10]	-	+	+	+	+	-

We are planning to create an individual page for each student. These pages would show the students' progress and provide AI-generated advice on each program so that each student can access to the information tailored for each of them and learn at any time. Students' feedback on the JC materials reveals that the cards are heavy. Currently, acrylic boards are used, which are durable for younger students. We need to explore alternative materials for the cards. Additionally, we are also considering modify the shapes of the cards related to "if," "else if," "else," and "while." This change aims to make the concepts of branching and iteration more intuitive.

## 6. Related Works

We referred to the literature on the development of tangible educational materials, literature on group learning analysis, and literature on education using Micro:bit, as listed below. Many related studies aim at the development of programming educational materials. Wang et al. developed and evaluated a programming-based maze escape game called "T-Maze" [5]. However, environments with multiple students like those in a school classroom setting were not taken into consideration. Tomohito Yashiro et al. developed a tool called "Plugramming" and conducted the construction and evaluation of a collaborative programming system using Scratch [6]. However, it fails to address situations where multiple students stumble at similar parts and encounter similar errors. Felix Hu et al. developed a tangible programming game called "Strawbies" for children aged 5 to 10 [7]. Programming is done using wooden tiles. Since the tiles are not square but have distinctive shapes, the users cannot make incorrect connections. Although the programming flexibility is reduced, it has the advantage of intuitively understandable whether a connection is possible or not. Aditya Mehrotra et al. proposed multiple approaches for programming education conducted in a classroom setting [8]. They utilized program blocks for robot programming and evaluated several methods. However, the evaluation was aimed at assessing the methods, and the system does not provide real-time instructions based on students' progresses. It does not promote knowledge retention either.

In many studies related to programming instruction, the main objective is to support programming. Koichi Kamichi designed a system for programming education without teaching assistants [9].

The system mirror learners answer to the server, providing automated suggestions of input errors for students and allowing monitoring of the progresses of the students. However, the automatic suggestions for input errors primarily aims to detect syntax errors, not considering programming novices who lack knowledge about logical thinking, which are prerequisites. Furthermore, the system only provides the instructor the number of errors that the student made, and does not provide more detailed analyses. Kato et al. developed a system in which they collected and analyzed the progress of students' programming in classes with teaching assistants and utilized this information effectively for teaching assistants so that they can guide students efficiently [10]. They conducted evaluation experiments demonstrating the system's effectiveness in instructional support. However, this analysis focuses on traditional programming languages and cannot be applied to tangible teaching materials.

Michail et al. systematically reviewed and summarized how the Micro: bit is used in primary education [11]. They reported that many students enjoy to use Micro:bit and found it easy to use. They evaluated it as beneficial for improving programming skills. In the survey, they demonstrated that Micro:bit is a promising tool for approaching STEM education. Dylan et al. conducted a two-week Micro:bit programming education program with 41 high school students [12]. After experiencing basic Micro:bit programming, the students became to be able to program autonomous cars equipped with Micro:bit and ultrasound distance sensors. Pre- and post-tests were conducted, and the results showed that the students' understanding of information processing and algorithms had deepened.

## 7. Conclusion

In this study, we reported our experiences of development of a classroom support system. This system assists students in programming and instructors who teach them. We conducted evaluation experiments to demonstrate the effectiveness of our system. We show a comparison of Jigsaw Coder (JC) with other related systems in Table 2. In general, it is difficult to monitor each student's progress in programming classes, and JC solves this issue. JC collects and analyzes each student's answer, and provides the instructor information for effective instruction. JC points out

program areas where many students are making mistakes in the class. This function helps the instructor to grasp the overall status of the class without inspecting students one by one. Furthermore, JC is a tangible learning system, and it allows students to learn programming through physical interaction. As long as a school can provide Micro:bits, paper QR cards, client PCs, a server PC, and a network, JC can be used in all economic regions around the world. Especially it is beneficial for students in developing countries. We conducted evaluation experiments of JC for high school students. They are new to programming. The students' responses were generally positive. The instructor's responses were also positive that JC could serve as an entry-level tool for programming. It allows the instructor to monitor the students' progress without moving around the classroom to check students one by one. Based on these results, we believe that JC is effective for programming education at a beginners level. On the other hand, authors are aware that the system has a serious limitation. We plan to revise the system with a parser and an analyzer to assist students building programming skills as well as logical thinking abilities. We reconsider the QR card design and try to make it simple too.

### Acknowledgment

This work was partially supported by JSPS KAKENHI Grant Number JP21K02805.

### References

- [1] O. Koji, K. Toshiyasu, Y. Kambayashi, "Development and Evaluation Experiment of a Classroom Support System for Programming Education Using Tangibles Educational Materials," Proceedings of the 12th International Conference on Information and Education Technology (ICIET), 67-71, 2024, DOI: [10.1109/ICIET60671.2024.10542715](https://doi.org/10.1109/ICIET60671.2024.10542715)
- [2] T. Kato, O. Koji, Y. Kambayashi, "A Proposal of Educational Programming Environment Using Tangible Materials," Human Systems Engineering and Design (IHSED2023), 1-8, 2023.
- [3] Y. Kambayashi, K. Furukawa, M. Takimoto, "Design of Tangible Programming Environment for Smartphones," HCI 2017: HCI International 2017, 448-453, 2017, DOI: [10.1007/978-3-319-58753-0\\_64](https://doi.org/10.1007/978-3-319-58753-0_64)
- [4] Y. Kambayashi, K. Tsukada, M. Takimoto, "Providing Recursive Functions to the Tangible Programming Environment for Smartphones," HCII 2019, 255-260, 2019, DOI: [10.1007/978-3-030-23525-3\\_33](https://doi.org/10.1007/978-3-030-23525-3_33)
- [5] D. Wang, C. Zhang, H. Wang, "T-Maze: A Tangible Programming Tool for Children," IDC '11: Proceedings of the 10th International Conference on Interaction Design and Children: 127-135, 2011, DOI: [10.1145/1999030.1999045](https://doi.org/10.1145/1999030.1999045)
- [6] T. Yashiro, K. Mukaiyama, Y. Harada, "Programming Tool and Activities for Experiencing Collaborative Design," Information Processing Society of Japan, **59(3)**: 822-833, 2018.
- [7] F. Hu, A. Zekelman, M. Horn, F. Judd, "Strawbies: Explorations in Tangible Programming," IDC '15: Proceedings of the 14th International Conference on Interaction Design and Children: 410-413, 2015, DOI: [10.1145/2771839.2771866](https://doi.org/10.1145/2771839.2771866)
- [8] A. Mehrotra, C. Giang, N. Duruz, J. Dedelley, A. Mussati, M. Skweres, F. Mondada, "Introducing a Paper-Based Programming Language for Computing Education in Classrooms," ITiCSE '20: Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education: 180-186, 2020, DOI: [10.1145/3341525.3387402](https://doi.org/10.1145/3341525.3387402)
- [9] K. Kamichi, "Designing a Programming Education Support System for Lessons without Practical Assistants," Journal of Sociology Research Institute, **1**: 73-78, 2020.
- [10] T. Kato, Y. Kambayashi, Y. Kodama, "Data Mining of Students' Behaviors in Programming Exercises," Smart Education and e-Learning, **59**: 121-133, 2016.
- [11] M. Kalogiannakis, E. Tzagaraki, S. Papadakis, "A Systematic Review of the Use of BBC Micro in Primary School," 10th International Conference New Perspectives in Science Education, STEM5036, 2021.
- [12] D. G. Kelly, P. Seeling, "Introducing Underrepresented High School Students to Software Engineering: Using the Micro Microcontroller to Program Connected Autonomous Cars," Computer Applications in Engineering Education, **28(3)**: 737-747, 2020, DOI: [10.1002/cae.22244](https://doi.org/10.1002/cae.22244)

**Copyright:** This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY-SA) license (<https://creativecommons.org/licenses/by-sa/4.0/>).