ICIC Express Letters
Part B: Applications
Volume **12**, Number **9**, September **2021**

ICIC International ©2021 ISSN 2185-2766
pp. **815–822**

A LEARNING SYSTEM FOR OBJECT-ORIENTED PROGRAMMING THEMED ON COLOR

Ryuki Moto, Daisuke Horii and Hiroshi Kamada*

Department of Media Informatics
College of Informatics and Human Communication
Kanazawa Institute of Technology
3-1 Yatsukaho, Hakusan, Ishikawa 924-0838, Japan
{ b1701847; b1701825 }@planet.kanazawa-it.ac.jp
*Corresponding author: kamada@neptune.kanazawa-it.ac.jp

Received March 2021; accepted May 2021

Abstract. *In recent years, the demand for object-oriented programming languages has been increasing year by year, and learning an object-oriented programming language is considered to be an indispensable concept for programming. In conventional systems, there was no learning system specialized in object-oriented programming that could display appropriate advice according to the result. Therefore, this study focused on learning an object-oriented programming language. We have increased the content of object-oriented programming with color-themed questions, and made it possible to display advice according to the answer results, making the system suitable for beginners. From the results of the evaluation experiment, it was confirmed that the usefulness of the system and the learning content were substantial.*
Keywords: Object-orientation, Programming language, Color, Learning system

1. **Introduction.** There are approximately 21.37 million IT engineers in 92 countries around the world [1]. Currently, there are about 1.09 million people in Japan, but the growth rate is slow compared to the United States and China. It is also sluggish in regions such as Latin America and Western Europe, and it is expected that the difference will widen in various regions as it is. In addition, in the language ranking that is attracting attention in the world, object-oriented programming languages occupy 16 languages (about 52%) among the top 20 languages, and this tendency is increasing year by year [2]. For this reason, learning an object-oriented programming language is an indispensable concept for programming and is considered inevitable. There are various types of conventional systems, such as a system that has a function of editing/adding implemented problems, a function of confirming terms, and a system that has a program input support function [3-5]. However, these systems do not have a function to display appropriate advice according to the result when answering the exercise. Therefore, it is necessary to find the wrong part by oneself, which is not suitable for beginners' learning. There is also [6] on prioritizing test cases for object-oriented languages, but the research is not applicable for beginners to learning object-oriented languages. There is also [7] on two-step learning of object-oriented programming by college students. The study states that it is important to learn object-oriented programming in a language that is easier to understand, starting with the basics. Therefore, there is a need for a system that can be used to develop human resources from beginner programmers, and that mainly targets object-oriented programming languages that are important for programming.

In the conventional system, there were problems such as the abstract object orientation and the inability to display appropriate advice according to the answer result. To solve

DOI: 10.24507/icicelb.12.09.815

these problems, we made it easier to learn abstract object-orientation with the concrete subject of color. We have also developed a new system that has never been seen before, with the ability to display appropriate advice according to the results and the implementation of two types of answer formats. As a result, we were able to increase learners' motivation to learn and reduce the difficulty of learning.

Conventional system and its problems are described in Section 2, the solution for the system in Section 3, the configuration of the system in Section 4, the evaluation experiment and its results in Section 5, the considerations described in Section 6, and we conclude in Section 7.

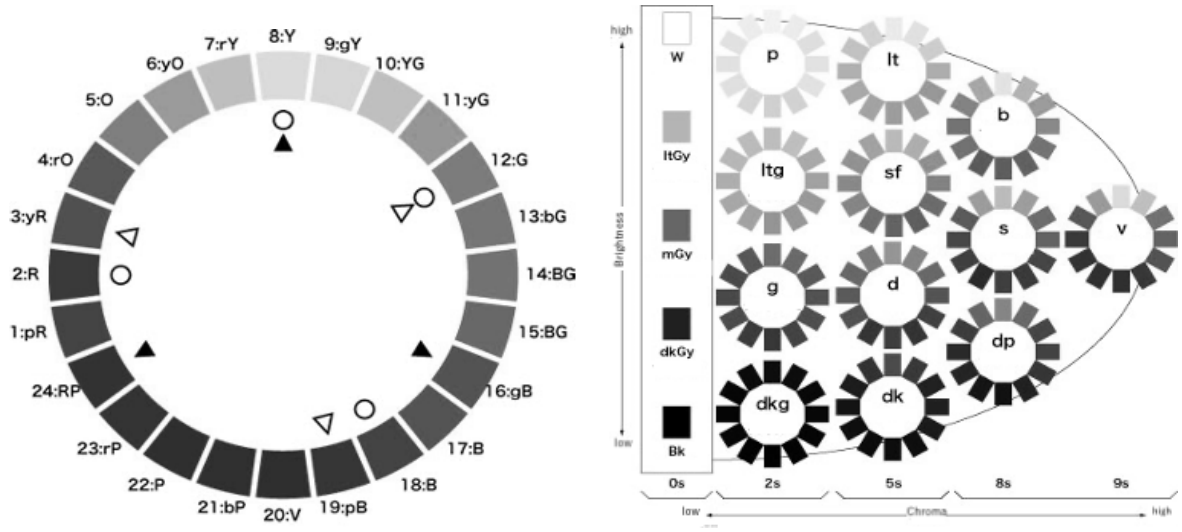
2. **Conventional System and Problems.** Conventional programming learning systems have various features such as a system that has a function to create and edit problems, a system that has a function to check terms, and a system that has a program input support function [3-5]. However, although you can learn basic functions and grammar with these systems, you cannot gradually deepen your understanding because the content of object-oriented programming, which is the basis of programming, is insufficient and exercises are difficult. From this, it can be said that there is no learning system for object-oriented programming for beginners. Also, the concept of object-oriented programming itself is abstract, which makes it difficult for beginners to understand. Therefore, it is difficult to apply what you have learned in the system to actual system development, and there is a problem that it is not suitable for beginners' learning.

The first conventional system [3] is a system called "QuizJET" announced by the University of Pittsburgh in the United States. The biggest feature of the system is that you can edit pre-implemented issues and add new ones. As a result, the educator or the like can ask the learner a new question, and can ask a question more suitable for the learner's understanding level. However, as a problem, in the exercises, there are no problem sentences and only the programs are listed, so if the learner is a programming beginner, it is necessary to think about what kind of program it is.

The second conventional system [4] is a system for learning the basic concepts of object-oriented programming through exercises using text materials, Flash videos, and illustrations. As a feature, it has a function to check terms such as textbooks, and it has a convenient function for beginners to start learning. However, as a problem, although the exercises are structured in a relatively easy-to-understand manner using illustrations, they are difficult for beginners because they are long and only descriptive formula. In addition, since there is no grade display, it is difficult for one person to proceed with learning efficiently.

The third conventional system [5] is a system for learning the basics of programming using a language called "xDNCL", which is an extension of the Japanese-based procedure description language DNCL. As a feature, the "program input support function" installed as an auxiliary function is very convenient. The "program input support function" is a function that even beginners who do not understand grammar can program in a fill-in-the-blank format. The problem is that there is no function to check terms, etc., and as with the conventional system [4], the exercises are in a descriptive format and there is no display of grades.

3. **The Solution for the System.** In the system, the abstract concept of object-oriented programming is facilitated by the concrete subject matter of hue and tone (saturation/lightness) of color. In this way, we aimed to solve the problem of the conventional system that "it is not suitable for beginners' learning". In addition, we used the Practical Color Co-ordinate System (PCCS [8]) color wheel and tone map to create color-related issues. This hue circle and tone map are shown in Figure 1.



(a) PCCS color wheel [8] (b) PCCS tone map [9]

FIGURE 1. PCCS color wheel and tone map

We aimed to solve the problems of the conventional system by pointing out the cause of the error and displaying the appropriate advice to give the correct answer for the answer including the typical error. This function advises typical misunderstandings by comparing the execution result image prepared in advance for each exercise with the answer result image of the learner. As a result, you can visually check the execution results of the questions answered during learning, and you can proceed with learning without problems.

4. **Configuration of the System.** The programming language used for learning is Processing, a simplified Java language. As a result, learning of object-oriented programming, which was not possible in conventional systems, can be focused on learning what object-oriented programming is through visual expressions such as figures and problems related to colors such as color wheel and tone. In addition, by dividing the problem into two stages, a choice format and a free description format, it was possible to develop not only basic knowledge but also applied skills. The system diagram of the learning items of the system is shown in Figure 2 below.

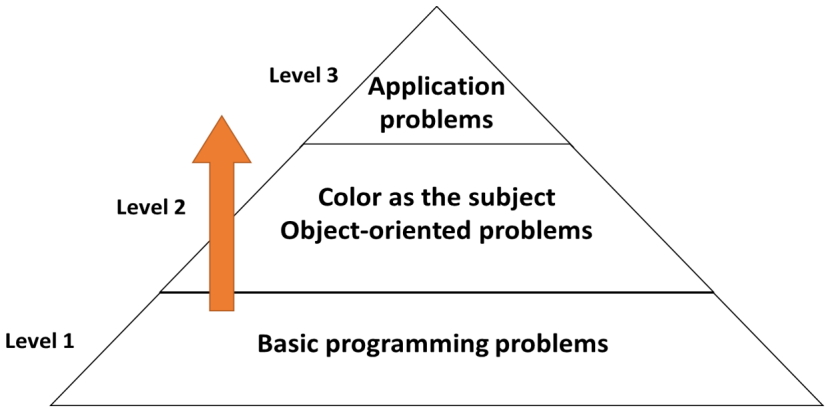


FIGURE 2. System diagram of the system

Also, the number of questions in the system is 82. Table 1 shows the learning content and the number of questions implemented in the system.

Since it is necessary for beginners to be able to learn from the basics to the application step by step, the problem structure is as described above. As a result, at Level 1, you can

TABLE 1. Learning content of the system

Level	Content of study	Learning items	The number of questions
3	Advanced edition (Application of object-oriented programming based on the introductory and intermediate editions)	Bounce ball	5
		Bouncing bubbles	4
2	Intermediate edition (Object-oriented programming based on color)	Hue problem	6
		Tone problem	6
1	Beginner's edition (Basics of programming)	Object-orientation	6
		For	5
		If	5
		Array	5
		Variable	10
		Paint	10
		Text	10
		Shape	10

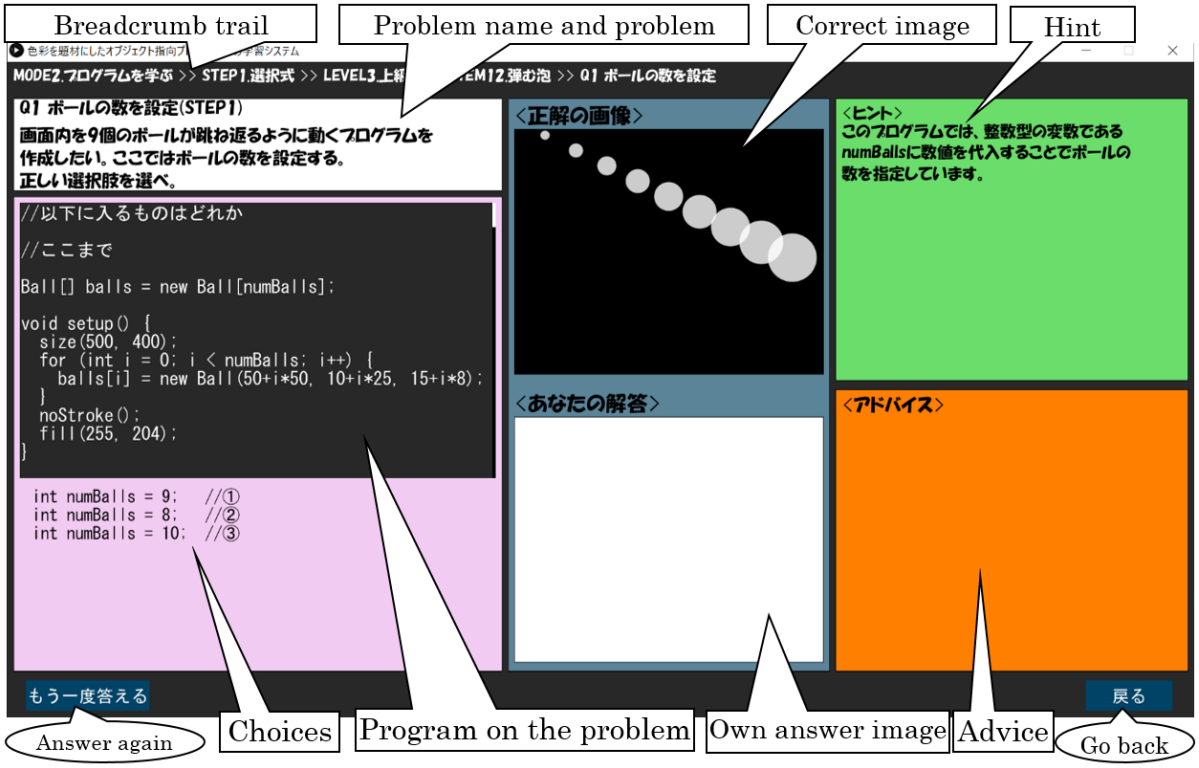


FIGURE 3. Example of answer screen

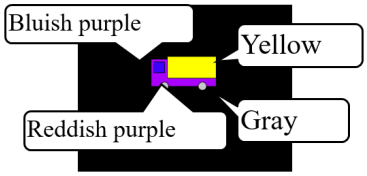
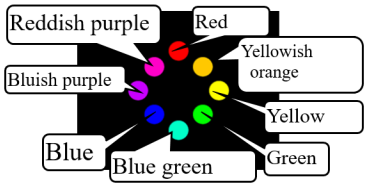
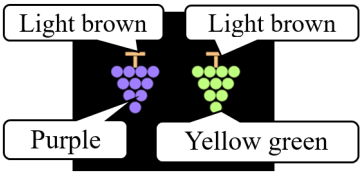
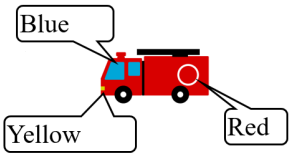
learn the most important knowledge for learning programming such as how to draw figures, how to draw characters, and conditional branching. At Level 2, you can learn more about object-oriented programming by using colors such as hues and tones as subjects. In the color-themed problem, in addition to figures such as spheres, we implemented problems related to color tones in two 2D models. Level 3 is an applied problem that combines a color problem with a moving program. This problem is completed by answering the problem step by step with reference to "The Nature of Code" [10], which is a part of the sample program of "Processing".

The screen when answering the system is as shown in Figure 3. At the top of the screen it is a breadcrumb trail that shows the path to the screen. The left side of the

screen is a question sentence and a program, and a text box for choice or free description is displayed below it. A scroll bar was implemented in the program part to support long programs. The place to answer is the part in the program between "//Which is below this" (Japanese) and "//So far" (Japanese). In addition, hints are always hidden in question sentences and programs in order to give correct answers to questions. The right side of the screen consists of the execution image at the time of correct answer and the answer image of the learner, the hint on the upper right, and the advice on the lower right. To answer the question, select an option that can display the same as the execution image at the time of the correct answer, or answer in a free description format. Figure 3 is an example of Q1 in the second learning item "Bouncing Bubbles" (Japanese) at Level 3. This is the question of answering the appropriate choices so that nine balls are displayed when the program is run. The hint for this problem says, "This program specifies the number of balls by assigning a number to the integer variable numBalls" (Japanese).

By pressing "Answer again" (Japanese) at the bottom left of the screen, the comment of the correctness judgment result displayed at the bottom right, "Own answer image" (Japanese), "Advice" (Japanese), and the saved image are deleted, and you can answer again. You can also return to the previous screen by clicking "Go back" (Japanese) displayed at the bottom right of each screen. The color-themed problem implements the problem of hue and tone. Regarding Level 2 color problems, Table 2 below shows an outline of the problem statement and two images to be used. The "Learning items" in Table 2 are the same as in Table 1. The "Number" represents the problem number in the system. In addition, "Problem overview" explains what kind of problem it is, not the problem statement, in an easy-to-understand manner.

Table 2. Outline of the problem statement and an example of the image used

Learning items	Number	Problem overview	Correct image
Hue problem	Q4	In answer to the program that draws the truck bed, create a truck with a combination of colors that will be the split complementary color scheme of yellow (ffff00).	
	Q6	In this issue, one color specification is missing from the pre-prepared program. Complete the color wheel by answering the program that creates the color.	
Tone problem	Q2	The program to create the grapes on the left and the shape of the grapes on the right are pre-specified. In this question, specify the color of the grapes on the right without changing the tones.	
	Q4	To create a fire engine with the same tones as car windows, headlights, and colors, derive the numbers from within the program and use the constructor to set the numbers.	

Regarding the saving of the execution result, the image of the result of the answer and execution by the learner is saved as an image of 500 × 400 pixels. Image judgment compares the saved 500 × 400 pixel image with the prepared 500 × 400 pixel image. In the conventional programming learning systems [11], all the problem data was managed in the program, but in order to simplify the program, the method was changed to read from the xls format Excel file.

Two types of answer formats were implemented: a choice format and a free description format. In the choice format, you can solve how to write a function while giving hints. Therefore, the difficulty level is lower than the free description format, and even beginners who have never touched the program can tackle the program problem just by inputting with the mouse. In addition, in the free description format, the same part as the choice format is described and solved by oneself to increase the difficulty level, and it is possible to step up to gain more practical skills.

In the case of the choice format, advice for correct/incorrect judgment is set for each selected answer. In the case of the free description format, five types of advice are set according to the answer of the answerer at the time of correct/incorrect judgment. As a result, even if the answerer writes an incorrect answer, advice is given according to the program entered. This advice function applied the function of the system in the submitted paper [12].

5. **Evaluation Experiment and Its Result.** In the evaluation experiment, an evaluation experiment was conducted on 29 students of Kanazawa Institute of Technology. After explaining the system using the web conferencing system "zoom", we asked them to answer the questionnaire. In the explanation, the system was executed, and one question each about hue and one question about color tone were answered from "Level 2. Intermediate". The answer format was a choice format and a free description. Six questions were prepared in the questionnaire, and each was answered on a five-point scale. The six question items were "Ease of use", "Usefulness", "Content and volume of problems", "Desire of continuous learning", "Difficulty" and "Reducing consciousness of weakness". The results are summarized in a graph like Figure 4.

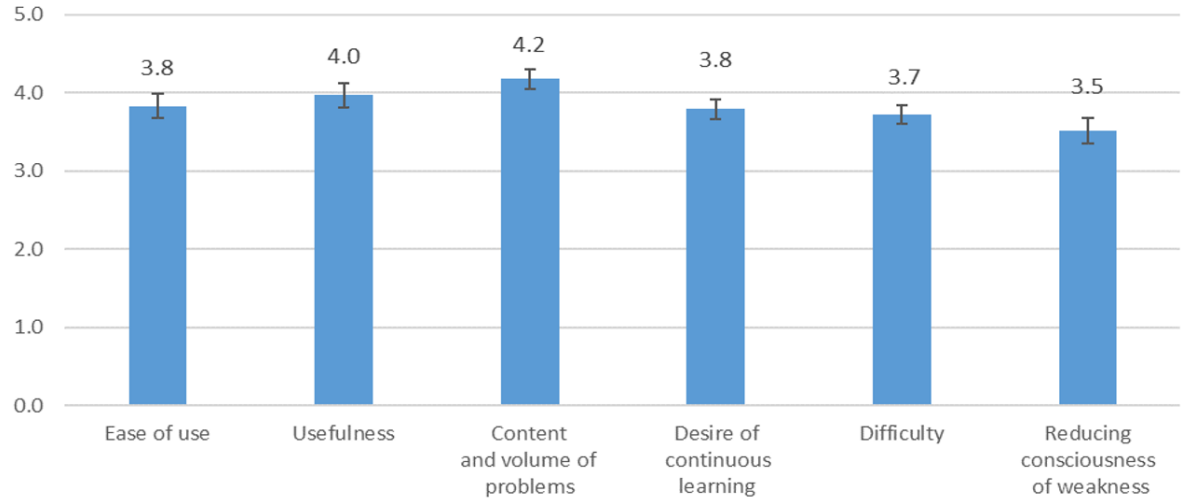


FIGURE 4. Results of evaluation experiment

"Content and volume of problems" has the highest evaluation at 4.2, and "Usefulness" has also the second highest evaluation result at 4.0. However, the "Difficulty" was 3.7, and the "Reducing consciousness of weakness" was 3.5, which was lower than the other questions.

6. **Consideration.** From the results of the evaluation experiment, the system was highly evaluated. The three items of "Ease of use", "Desire of continuous learning", and "Difficulty" were evaluated on average, but the item of "Content and volume of problems" was the highest evaluation result. The item of "Reducing consciousness of weakness" was evaluated lower than others. We think from these results, we introduced as well as a glossary, a score confirmation that can confirm the saved answer, and a still image problem, and a dynamic problem so the average value of "Usefulness" and "Motivation for continuous learning" increased. However, in terms of "Ease of use", the difficulty level was suitable for beginners, but we think it is difficult to use because it takes time to add multiple libraries, start up slowly, and make correct and incorrect decisions.

From the experimental results of the evaluation questionnaire, it can be seen that the "Content and volume of problems" of the system received high evaluation in all items. From this result, it is considered that it was evaluated that a sufficient number of problems were implemented for the user to learn. However, it can be seen that "Difficulty" and "Reducing consciousness of weakness" were evaluated lower than other items. From this result, it is considered that the beginners were given the impression that there were few simple problems and they could not enjoy learning programming. In the future, it will be necessary to devise ways to maintain learning motivation, such as implementing problems that are more interesting to learners.

As a novelty, when the learner's answer is not correct, it is collated with a typical wrong answer and appropriate advice is presented to the learner's wrong answer. As a result, the learner's misunderstanding can be pointed out, and it becomes easier to learn by the system alone. However, the poor usability is an issue for maintaining the learner's motivation to learn.

7. **Conclusion.** We have developed the system to learn an object-oriented programming language. Among them, there were problems such as the abstractness of object-oriented programming, insufficient learning content for object-oriented programming, and the inability to display appropriate advice according to the results when answering exercises. To solve these problems, we made it easier to learn abstract object-orientation by using concrete colors as the subject matter. In addition, the number of questions was increased for the parts where the content was insufficient, and questions with a wide range of difficulty from 1 to 3 were implemented. From the results of the evaluation experiment, it can be said that these problems have been solved.

By displaying appropriate advice for the answer to the question, it is possible to point out what is different even if the answer is incorrect, and the learner's motivation for learning can be improved. Also, if you cannot check your grades, you can save your own answer by implementing a score check function. The difficulty of the exercises is that we have prepared a selection format and a free description format so that the learner can learn in a suitable way. Looking at the results of the evaluation experiment, it can be said that these problems have been solved to some extent.

It can be said that the system was able to develop a system suitable for beginners' learning from the implementation and evaluation results of the advice display function for learners and the score confirmation function.

In the future, it will be necessary to make improvements of the system such as implementing more detailed advice functions.

Acknowledgment. This work was supported by JSPS KAKENHI Grant No. 18K02836.

REFERENCES

[1] Human Resocia Co., Ltd., *[Human Resocia Research] Global IT Engineer Report vol.1 92 Countries*, https://git.resocia.jp/en/info/post-developers-around-the-globe-survey/, Accessed on February 15, 2021.

[2] TIOBE Software BV, *TIOBE Index for February 2021*, https://www.tiobe.com/tiobe-index/, Accessed on February 2, 2021.

[3] I.-H. Hsiao, S. Sosnovsky and P. Brusilovsky, Guiding students to the right questions: Adaptive navigation support in an E-Learning system for Java programming, *Journal of Computer Assisted Learning*, pp.270-283, 2010.

[4] M. Ohshiro and Y. Nagai, The proposition of an object-oriented programming education system for programming learners: Focused on making and executing programs based on understanding basics of object-oriented, *Information Education Symposium 2016 Proceedings*, pp.114-121, 2016 (in Japanese).

[5] T. Nishida, A. Harada, R. Nakamura, Y. Miyamoto and T. Matsuura, Implementation and evaluation of PEN: The programming environment for novices, *Transactions of Information Processing Society of Japan*, pp.2736-2747, 2007 (in Japanese).

[6] Vedpal and N. Chauhan, Test case prioritization technique for object oriented software using method complexity, *International Journal of Innovative Computing, Information and Control*, vol.14, no.1, pp.341-354, 2018.

[7] T. A. Dmitrieva, A. V. Prutzkow and A. N. Pylkin, Two-level study of object-oriented programming by university students, *Modern Information Technologies and IT Education*, vol.15, no.1, pp.200-206, 2019.

[8] Japan Color Enterprise Co., Ltd., *PCCS Hue*, http://www.sikiken.co.jp/pccs/pccs02.html, Accessed on March 8, 2021 (in Japanese).

[9] Japan Color Enterprise Co., Ltd., *PCCS Tone*, http://www.sikiken.co.jp/pccs/pccs04.html, Accessed on March 8, 2021 (in Japanese).

[10] D. Shiffman, *The Nature of Code: Simulating Natural Systems with Processing*, Magic Book Project, 2012.

[11] S. Sasaki, H. Takahashi, K. Takeda and H. Kamada, Systematic learning system for programming from the basics, *2019 PC Conference Proceedings*, pp.208-211, 2019 (in Japanese).

[12] S. Sasaki and H. Kamada, A description-type programming learning system which advises about main wrong answers, *ICIC Express Letters, Part B: Applications*, vol.12, no.9, pp.807-813, 2021.