Relationship Analysis between Puzzle-Like Programming Game and Achievement Result After Learning the Basic of Programming

Shimpei Matsumoto, *Member, IAENG*, Shuichi Yamagishi and Tomoko Kashima, *Member, IAENG*

***Abstract*—Japan Electronics and Information Technology Industries Association develops a puzzle-like programming game "Algologic" aiming to experience the concept of an algorithm for inexperienced programming learners. This is simple puzzle game aiming at solving a problem by automatically controlling a robot. The user designs an autonomous robot by selecting some of the instruction blocks, arranging the blocks in an appropriate order, and giving them to the robot. Before university students learn the basic of programming, with this programming puzzle game, we conducted a test to determine whether algorithms each student gave to the robot were correct or not. Likewise, after students have learned the basic of programming, we conducted a comprehension test to clarify the reachability. In this paper, we aim to investigate the relation between the comprehension of Algologic and the achievement degree of students after learning of programming and report the analysis result. Analysis results revealed that the results of Algologic test and the achievement results after learning programming were significantly in a positive relationship.**

***Index Terms*—programming game, puzzle game, basic programming skill, learning analytics.**

I. Introduction

USUALLY, introduction education for programming beginners aims three primary goals: conveying the concept of programming, conveying the fun of programming, and giving learners experience with important things for learning programming. As an approach to realize these goals, puzzle-like programming games or visual programming languages have been actively adopted in programming introduction education. The difference between programming game and visual programming is whether the goal is given in advance or not. In general, programming games provide a definite goal and constrain problem-solving means. On the other hand, visual programming only provides a means to make programming easier, and its learners themselves can set goals freely. Programming games are easy to use for a lesson because its goal is clear, so they have often been used to convey the essence of programming in a short time.

As one of the famous Japanese programming games, the Japan Electronics and Information Technology Industries Association has developed a web application named "Algologic". Algologic is an algorithm experience puzzle game for inexperienced programmers, and the main target of this game is junior high school and high school students who are interested in computer science and want to learn the details. The authors have assumed that getting used to thinking algorithm has a good influence on programming learning, so have utilized Algologic at the early stages of a programming introduction class at university. Also, to check the progress of the exercise, the authors have carried out an Algoligic featuring test to grasp the learner's comprehension degree and overall tendency in the lecture right after the exercise with Algologic in this class. There are many practical examples using programming games for programming education, but there are few efforts to gather data on learning activities when using programming games and to utilize them for the analysis of programming learners. At the stage when learners who are willing to master programming just started programming learning, investigating what kind of learning activities were carried out in programming game and analyzing the relationship between the score of programming game with the achievement result after learning the basics of programming are considered to be valuable works. Such research will provide useful data for enhancing programming education such as the design of instructional method and lecture planning, estimation of difficulty in an algorithm, and extraction of learners who may have difficulty keeping up with the lecture.

Then, this paper aims to investigate the relationship between the score of programming game and the achievement degree of an actual programming lecture by using Algologic as a programming game and reports the analysis results. As mentioned above, before students touch programming, the authors tested whether each student can find an appropriate algorithm to Algologic featured questions and collected data that can be used to evaluate the essential skill to learn computer science. With the collected data, the authors grasped the pre-state of each student. After students learned all critical programming contents, the authors performed an achievement test to clarify the post-state of each student. By using the above both data, this paper conducted a pre- and post-survey on each student's programming learning. Specifically, the authors examined what kind of result students obtain in the programming game according to each student's achievement result. The analysis result in this paper showed that the correct answer rate of programming game and the achievement result of programming are a positive relationship. This finding suggests that students who make programming learning difficult may not have enough computational thinking skill [1]-[3] from the from the previous point before programming learning.

Shimpei Matsumoto and Shuichi Yamagishi are with the Department of Information Systems and Management, Faculty of Applied Information Science, Hiroshima Institute of Technology, and also with the Graduate School of Engineering. Address: 2-1-1 Miyake, Saeki-ku, Hiroshima 731-5193, Japan. e-mail: {s.matsumoto.gk, s.yamagishi.if}@cc.it-hiroshima.ac.jp. URL: http://www.it-hiroshima.ac.jp/

Tomoko Kashima is with the Department of Information and Systems Engineering, Faculty of Engineering, Kindai University. Address: 1 Takaya Umenobe, Higashi-Hiroshima City, Hiroshima, 739-2116, Japan. E-mail: kashima@hiro.kindai.ac.jp

II. Related Works

In recent years, expectations and requests for programming education have increased more than ever. Therefore, various programming games have been developed and used in the lessons at colleges or private schools. RoboCode has been famous for a long time, and Minecraft and Elevator Saga are drawing much attention recently [1]. For the same reason, development and research of visual programming languages are also advanced [4]-[8].

Squeak [9] and Scratch [10]-[14] are representatives of visual programming languages. Since visual programming languages can avoid grammatical errors, they make programming learners concentrate on building and comprehending algorithms. Therefore, it would be an easy-to-use programming language for beginners. In particular, visual programming languages are very effective for programming learning of non-English speaking children. There are efforts to train the programming skill and the ability to build algorithms at the same time for mainly college students, and the representative examples are BlockEditor[15] and oPEN[2][16]. These are using a visual programming library named OpenBlocks [18] and realize a smooth transition from the conventional block type language to the text type language.

Whether programming games are effective to train programming skills or not is unclear at this time. Hour of Code [3] rejects that many of the programming games are not developed for the purpose of enhancing programming skills, creativity, problem-solving skill. In other words, general programming games aim to introduce the players that learning computer science is close, useful, simple and exciting. Several examples of using programming games for programming education have been done [18], but it seems to be practiced under the same policy as Hour of Code.

III. Skill Check Test Using Programming Game

We can find many studies on programming education showing the effectiveness of introducing a programming game from the learners' opinions and comments collected by a questionnaire. However, there have been few studies on analyzing the degree of understanding of programming games themselves and also few case studies on considering programming achievement degree with the score of programming game. Therefore, the authors focus on clarifying the relationship between the score of programming game and the achievement degree of basic programming class for college students.

A. Programming Game "Algologic"

Algologic is a puzzle-like game, and its user aims the goal by automatically controlling a robot by predetermined instruction blocks. There are two types of goals: the one requires to collect all flags set in the field, and the other requires to control the robot along with the indicated directions. For each question, the player selects some of the instruction blocks prepared in advance, arranges the blocks in the appropriate order, gives it to the robot, and automatically

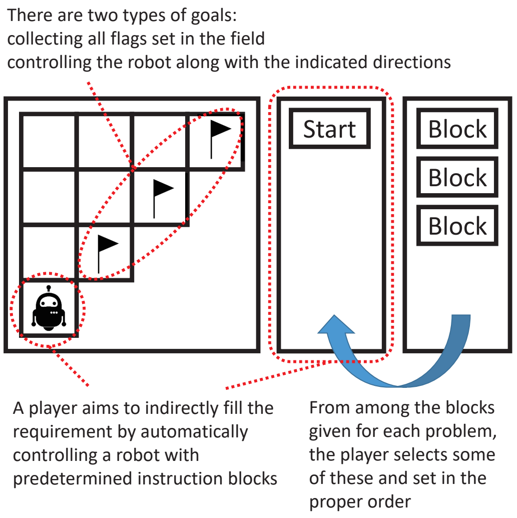
There are two types of goals:
collecting all flags set in the field
controlling the robot along with the indicated directions
Start
Block
Block
Block
A player aims to indirectly fill the requirement by automatically controlling a robot with predetermined instruction blocks
From among the blocks given for each problem, the player selects some of these and set in the proper order

Fig. 1. The screen structure of Algologic

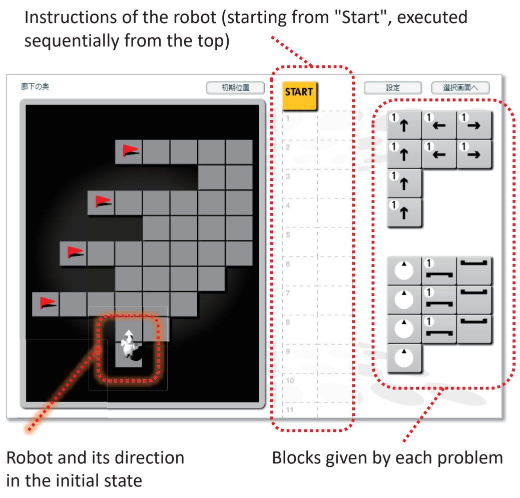
Instructions of the robot (starting from "Start", executed sequentially from the top)
START
Robot and its direction in the initial state
Blocks given by each problem

Fig. 2. The actual operation example of Algologic requiring to collect all flags set in the field

controls the robot. The player can clear the game if the robot indirectly satisfies the problem's needs. The screen structure of Algologic and its actual operation example are shown in Fig. 2 and Fig. 3 respectively. Some explanations of Algologic are below.

- The player connects an instruction block under the start block by dragging and dropping. Algologic provides instructions, front, back, left, right, rotation, repeat, and branch (only Algologic 2) as the instruction block. The player can set the amount of movement and the number of repetitions by an integer value from 1 to X to the instruction block.
- When the player clicks the start button after arranging the instruction blocks, the robot moves in accordance with the given instruction blocks. If the start button is clicked while the robot is moving, the robot pauses.
- When the player clears with the smallest instruction block, The message that the algorithm is the best solution is displayed on the screen. The player can clear a stage without an optimal solution, but cannot receive the message showing the best solution.

[1]http://www.businessinsider.com/15-free-games-that-will-help-you-learn-how-to-code-2017-4/
[2]https://github.com/xDNCL/oPEN
[3]https://hourofcode.com/us

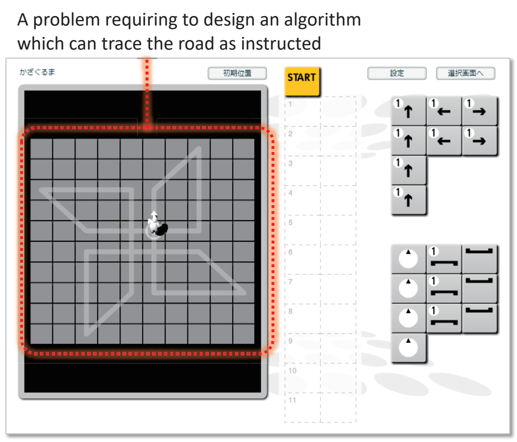
A problem requiring to design an algorithm which can trace the road as instructed
START

Fig. 3. The actual operation example of Algologic requiring to control the robot along with the indicated directions

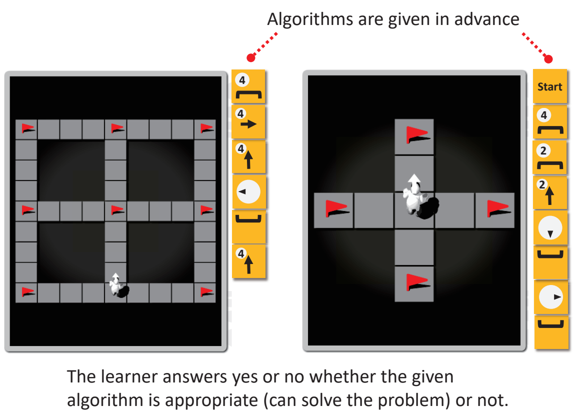
Algorithms are given in advance
Start
The learner answers yes or no whether the given algorithm is appropriate (can solve the problem) or not.

Fig. 4. An example of a question in the comprehension check test

B. *Instruction with Algologic*

College students have learned what is the essence of programming by using Algologic at the beginning of a basic programming class. Students took Algologic exercises in two 90 minute lectures. The authors checked individually whether each student understood the rule of Algologic properly, and explained the rule over and over again if his/her understanding is inadequate. After confirming that all students understood the rules enough, the authors conducted a test to confirm their comprehension using Algoligic featuring test. This test does not require students to design and construct an algorithm from scratch, but ask to answer by "yes " or "not" whether a given algorithm for each field is suitable or not. An example of a question is shown in Fig. 4. Specifically, using Moodle's Quiz module, each student gave his/her answer with a two-alternative form to each question "the game is clear if the robot can take all the flags / if the robot can trace the road as instructed. Answer whether the algorithm on the right side is appropriate to solve the problem or not. It is not necessary that the algorithm does not have to be optimal". The time limit of the test is 10 minutes, and the number of questions is 10. The test did not allow talking each other and bringing personal belongings to obtain accurate data as much as possible.

IV. Analysis Results

The analysis target of this paper was 1st-grade college students majoring informatics with little programming experience. The authors collected learning log data of basic programming classes over two years, analyzed learners' data $n = 124$ in the 1st year's classes and $n = 109$ in the next year. These data excluded second and higher grades students and also students who declined the programming class on the way. Students took the Algologic test at the beginning of basic programming class for 1st-grade students held in the 1st term. Similarly, there was a full-fledged programming class for 1st-grade students held in the 2nd term, and this paper used the score of this class's examination result for analysis. These lectures adopted C language to programming. For the analysis of the relationship between before and after programming learning, this paper used learner's data who took both 1st and 2nd term classes. Therefore, the analysis targets were $n = 113$ in the 1st year's class and $n = 99$ for the next year.

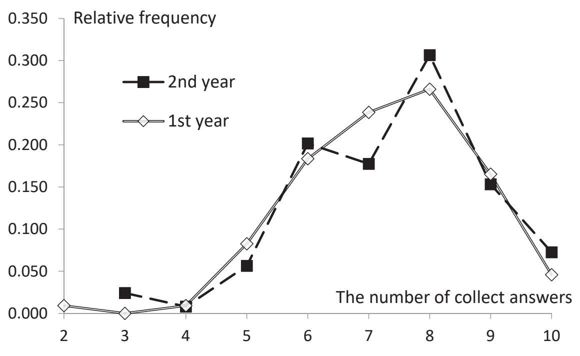

Fig. 5. Histogram of Algologic tests

The same lecturer was responsible for both the 1st and next year's classes, and the content of the lesson and also the difficulty level of the Algologic test were comparable. In the 1st year, the average GPA of the 1st term class was 2.12, and the 2nd term class was 2.35. Similarly, in the next year, the average GPA of the 1st term class was 2.03, and the 2nd term was 2.22. Although the results of the next year were lower than the 1st year, there was no significant difference between the averages of GPA by Welch's t-test. Therefore, the programming ability of each group can be regarded as almost the same degree.

Fig. 5 shows histograms of the score of Algologic tests. The vertical axis is the relative frequency, and the horizontal axis is the score of the test where its perfect score is 10 points. There was no significant difference between the average scores where the first year was 7.38, and the next year was 7.31. As shown in Fig. 5, both histograms of the two years were similar trends. Therefore, based on the values of average GPA mentioned above, two year's variations of academic level are considered to be almost same. From this histogram, it became clear that about 30% of learners could not obtain 7 points or over. This test did not require designing and building algorithms, but instead only predicting the behavior as a processing result by interpreting the instructions in order. Designing and constructing an algorithm is an essential task of programming before coding source code. Therefore, the result of Fig. 5 suggests that about 30% of learners did not have enough knowledge necessary for advancing intrinsic learning of programming.

This paper analyzed the relationship between the achievement result after learning programming and Algologic test.

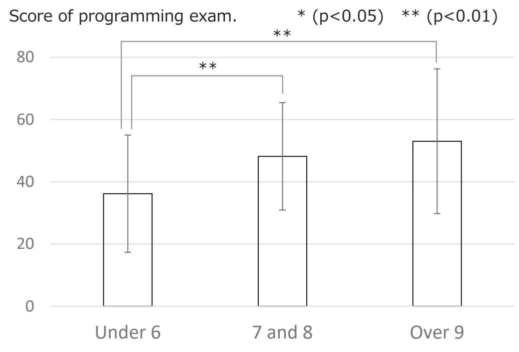

Fig. 6. Relationship between the score of programming examination in 2nd term and the score of Algologic test (1st year's data)

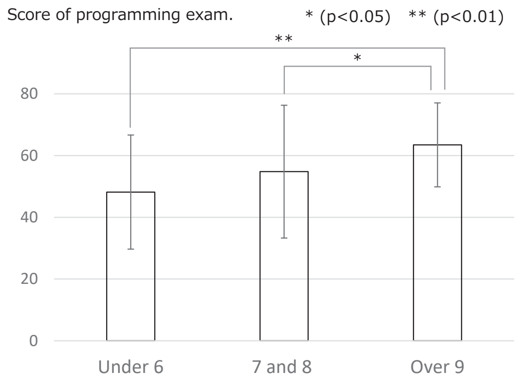

Fig. 7. Relationship between the score of programming examination in 2nd term and the score of Algologic test (2nd year's data)

As the programming class in the 2nd term gave students the achievement verification test of 100 points twice, so the average of the two scores was used as the achievement result. Fig. 6 and Fig. 7 shows the analysis results where the vertical axis is the average of the achievement result, the horizontal axis is the score of Algologic test, and error bars represent standard deviation. In Fig. 6 and Fig. 7, all students divided into three groups according to the score of Algologic test. As shown in Fig. 6 and Fig. 7, there was a positive relationship between Algologic test and the achievement result, and there were significant differences among several items by Welch's t-test. There are several possibilities as a reason for showing the positive relationship. The first possibility is that some kind of sense is indispensable to learning programming [19], and a learner with no sense could not acquire the knowledge. Next possibility is that a learner stumbled at the early stage of programming class could not learn the concept of programming smoothly. In any case, the ability to properly grasp an algorithm / a procedure of instructions and imagine an output for a given input would be more important programming learning than memorizing programming language specification and coding a source code. On the other hand, some students did not have a good achievement result of programming even though the score of Algologic test is good or vice versa. Since the analysis result of this paper was only from a global perspective, the authors will perform a further detailed analysis, for example, classification of students under various conditions, or student's pattern analysis based on the contents of questions.

V. Conclusion

This paper investigated the relationship between the score of the programming game and the achievement result of actual programming students by using Algologic and reported the analysis results. The analysis result in this paper showed that the correct answer rate of Algologic and the achievement result after learning the basics of programming were a positive relationship.

Acknowledgments

This work was partly supported by Japan Society for the Promotion of Science, KAKENHI Grant-in-Aid for Scientific Research(C), No.16K01147 and No.17K01164.

References

[1] J. Wing, "Computational thinking," *Communications of the ACM*, 49, 3, pp. 33-35, 2006.
[2] K. Brennan, M. Resnick, "New frameworks for studying and assessing the development of computational thinking," *In AERA 2012*, 2012
[3] Google, "Google's Exploring Computational Thinking," http://www.google.com/edu/computational-thinking/, Accessed 16 Nov. 2017.
[4] S. Cooper, W. Dann, R. Pausch, "Teaching objects-first in introductory computer science," *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education*, pp. 191-195, 2003.
[5] J. C. Cheung, G. Ngai, S. C. Chan, W. W. Lau, "Filling the gap in programming instruction: a text-enhanced graphical programming environment for junior high students," *Proceedings of the 40th ACM Technical Symposium on Computer Science Education*, 2009.
[6] E. Pasternak, "Visual Programming Pedagogies and Integrating Current Visual Programming Language Features," *Master's Thesis*, Carnegie Mellon University, Robotics Institute Master's Degree, 2009.
[7] A. Warth, T. Yamamiya, Y. Ohshima, W. Scott, "Toward a More Scalable End-user scripting Language," *Proceedings of 2nd International Conference on Creating Connecting and Collaborating through Computing 2008*, pp.172-178, 2008.
[8] Google Inc., "Blockly: a visual programming editor," https://developers.google.com/blockly/, Accessed 16 Nov. 2017.
[9] D. Ingalls, T. Kaehler, J. Maloney, S. Wallace, A. Kay, "Back to the future: the story of squeak, a practical smalltalk writtern in itself," *Proceedings of ACM OOPSLA 1997*, pp. 318, 1997.
[10] M. Fal, N. Cagiltay, "How scratch programming may enrich engineering education," *Proceedings of 2nd International Engineering Education Conference*, pp. 107-113, 2012.
[11] J. Maloney, L. Burd, Y. Kafai, N. Rusk, B. Silverman, M. Resnick, "Scratch: a sneak preview," *Proceedings of 2nd International Conference on Creating Connecting and Collaborating Through Computing*, pp. 104-109, 2004.
[12] C. Lewis, "How programming environment shapes perception, learning and goals: logo vs. scratch," *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, pp. 346-350, 2010.
[13] D. Ozoran, N. Cagiltay, D. Topalli, "Using scratch in introduction to programming course for engineering students," *Proceedings of 2nd International Engineering Education Conference*, pp. 125-132, 2012.
[14] Scratch Team Lifelong Kindergarten Group MIT Media Lab, "Scratch -imagine.program.share-," https://scratch.mit.edu, Accessed 16 Nov. 2017.
[15] Y. Matsuzawa, Y. Tanaka, S. Sakai, "Measuring an Impact of Block-Based Language in Introductory Programming," T. Brinda, N. Mavengere, I. Haukijarvi, C. Lewin, D. Passey (eds), *Stakeholders and Information Technology in Education, SaITE 2016, IFIP Advances in Information and Communication Technology*, vol. 493, Springer, 2016.
[16] T. Nishida, A. Harada, T. Yoshida, R. Nakamura, et al., "PEN: A Programming Environment for Novices - Overview and Practical Lessons -," *EdMedia: World Conference on Educational Media and Technology*, pp. 4755-4760, 2008.
[17] R. V. Roque, "OpenBlocks An Extendable Framework for Graphical Block Programming Systems," *Electrical Engineering and Computer Sciences - Master's degree*, 2007.
[18] T. Saga, "Learning programming with algo logic and programin,", *The Journal of Wakkanai Hokuseigakuen College*, Vol. 12, pp.99-111, 2012, In Japanese.
[19] R. Bornat, "Camels and humps: a retraction," Middlesex University, http://www.eis.mdx.ac.uk/staff pages/r_bornat/papers/, Accessed 16 Nov. 2017.