Analysis of the Learning Effects between Text-based and Visual-based Beginner Programming Environments

Authors Name/s per 1st Affiliation (*Author*)
line 1 (of *Affiliation*): dept. name of organization
line 2-name of organization, acronyms acceptable
line 3-City, Country
line 4-e-mail address if desired

Authors Name/s per 2nd Affiliation (*Author*)
line 1 (of *Affiliation*): dept. name of organization
line 2-name of organization, acronyms acceptable
line 3-City, Country
line 4-e-mail address if desired

***Abstract*—Although different programming environments have been developed to teach programming to beginners, the programming environment most suitable for introductory education is unknown in terms of the programming method such as text or visual. This study applies programming learning using a digital game, Minecraft, to compare the learning effect of text-based and visual-based programming in the LUA programming environment provided by a Minecraft extension. Both methods have certain learning effects, but visual-based programming has more significance than text-based programming on the attitude toward programming, indicating that visual-based programming is superior to text-based programming in introductory education.**

Keywords—Programming language learning; Introductory education; Digital game-based learning; Minecraft

I. Introduction

Both visual and textual environments have been used to teach programming to beginners. Examples of visual methods include Scratch [1] and Alice [2], while Java and Python are examples of textual methods. In some studies, visual methods are more suitable for beginners [3, 4, 5, 6]. However, beginners can also learn to program from a text language [13]. Whether one programming method (text or visual) is more suitable for introductory education remains unknown. Herein the difference between the two (visual and text) input methods is investigated to elucidate if the environment influences the learning effect. Specifically, this research examines the following research questions:

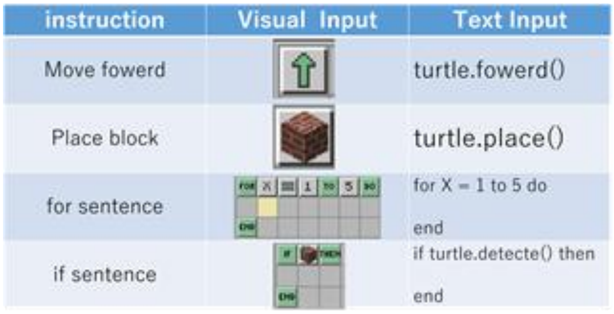
- RQ1: Do visual-based programming and text-based programming induce different attitudes toward programming?
- RQ2: Does the understanding of programming basics vary between visual-based and text-based programming?
- RQ3: Does the understanding of programming concepts differ between visual-based and text-based programming?

RQ1 assesses whether the programming method is suitable as an introductory environment by assessing learners' attitudes toward programming. RQ2 evaluates the understanding of programming basics with an emphasis on how to use the functions and inputs. RQ3 examines the understanding of programming concepts by focusing on sequential execution, conditional branching, and repetition. These RQs should help determine the appropriate programming method and environment for introductory education because increasing the learning efficiency should enhance the learning effect. In addition, the proper learning environment should improve beginners' motivation to learn.

We used Minecraft [7] as a learn-to-program platform. Minecraft is the most popular sandbox game in the world, and is used in education, including programming education. In addition, we also used ComputerCraft (CC) [8], which is a mod that adds the function of LUA to the programming language and the turtle programming environment in Minecraft. CC can control turtle programming using LUA. This extension is ComputerCraftEdu (CCEdu) [9] of the education version. CCEdu provides two environments (Visual and Text) (Fig. 1). Text-based programming can be controlled in Minecraft using the same method as general text-based programming, while visual-based programming employs visual-blocks. Both environments have the same level of abstraction. For example, the instruction 'turtle.forward()' moves the turtle forward. Figure 2 shows the relationship between the two methods.

In this paper, we explain background of the study in chapter 2. Chapter 3 discusses the Lecture Design in order to compare the two methods. Chapter 4 explains the experiment using the Lecture Design in Chapter 3. Chapter 5 discusses the results and analyses of the experiment. Chapter 6 discusses the results and threats to the validity. Chapter 7 shows related works, while Chapter 8 is the conclusion and future work.

Fig. 1. Two programming environments in CCEdu



instruction	Visual Input	Text Input
Move foward		turtle.foward()
Place block		turtle.place()
for sentence		for X = 1 to 5 do end
if sentence		if turtle.detecte() then end

Fig. 2. Relationship between the two input methods

II. BACKGROUND

A. *Visual and Text Programming Methods*

Visual-based programming (e.g., Scratch and Alice) employs a drag-and drop feature, which increases programming ease. Visual-based programming is appropriate for programming beginners [3, 4, 5, 6]. However, text-based programming (e.g., C and Java) is also applicable when first learning to program [14]. In addition, [10] compared a text environment to a visual environment using basic functions of a block interface in a text environment. In contrast, we have investigated the more suitable programming environment (visual or text) for introductory education using the same level of abstraction. For example, CCEdu has both visual and text methods in the environment for the LUA language, which is a text language. We investigated the input method in same language.

TABLE I. RESEARCH SCOPE OF PROGRAMMING ENVIRONMENTS(VISUAL AND TEXT)

	Text-based input	Visual-based input
Text-based language	CCEdu (Text)	CCEdu (Visual)
Visual-based language		Alice, Scratch

B. *Programming education with Minecraft*

A previous study investigated programming learning in a workshop using CC, text programming and a Minecraft [11]. Employing CC improves learning motivation. Reference [12], which compared text-based and visual-based programming education, found that these methods have similar relationships to programming languages, but their coding environments differ. Although both methods (text and visual) increase motivation, which method is better suited to educate beginning programmers has yet to be revealed. Thus, we investigate the effects of the programming method on learning by implementing two different methods using the same format and level of abstraction in the same environment.

III. LECTURE DESIGN

We implemented two types of lectures (visual-based and text-based). Each lecture was designed for elementary and junior high school students in Japan. Both covered the same material using a short course for introductory programming. This educational goal of the course was to teach the concepts and basics of programming. Each lecture consisted of a tutorial, sequential execution, repeat, conditional branching, and a free problem. The tutorial content focused on operations in Minecraft and CCEdu, while sequential execution involved learning a sequential execution, which is a programming basic. The example in the lecture was to move a turtle (forward, back, up, and down) and place a block in Minecraft. We taught the instructions of the turtle, which can move forward, back, left, right, up, or down. Additionally, the concept of repeat was explained using the “for sentence” to place blocks (Stack and Load Line) with the turtle. Examples included stacking five blocks and creating a staircase pattern. On the other hand, Conditional branching was explained using the “if sentence” to avoid a block. The lecture used two examples: “avoid obstacles” and “remove TNT.”. Finally, a final challenge, which was to create the alphabet letter of the first letter in their name, was used to assess the students’ programming skills. In addition, to gauge the understanding of programming basics, the lecture included six problems (TABLE II).

The total time of the lecture course was approximately 3.5 hours, which was allocated as follows: Tutorial (30 minutes), Sequential execution (50 minutes), Repeat (25 minutes), Conditional branching (25 minutes), Free problems (30 minutes), and a Break (30 minutes). Although the course was short, the programming concepts of conditional branching, repeat, and sequential execution were taught.

TABLE II. LIST OF PROBLEMS

Problem No	Problem	Contents
P1	Move the turtle three steps, rotate left, and move two more steps.	Sequential execution
P2	Add four blocks.	Sequential execution
P3	Stack eight blocks.	Repeat
P4	Create a stairway with eight steps.	Repeat
P5	If a TNT block is in front of the turtle, avoid it.	Conditional branching
P6	If a diamond block is in front of the turtle, mine it.	Conditional branching

IV. EXPERIMENT

We confirmed whether the text or visual method is better suitable for introductory education by comparative experiments based on the "Lecture Design" described in the previous chapter. In addition, we developed three hypotheses that correspond to the RQs:

- H1: The visual-programming lecture induces a larger change in attitude toward programming. (RQ1)
- H2: Programming basics are easier to understand using visual-based programming. (RQ2)
- H3: Programming concepts are easier to grasp using visual-based programming. (RQ3)

H1 is the RQ1 hypothesis. It is speculated that the visual-based group will have a more significant change in attitude toward programming because visual-based programming is more intuitive than text-based programming. H2 is the RQ2 hypothesis. Similar to the rationale for H1, it should be easier to comprehend programming basics using illustration-based programming. H3 is the RQ3 hypothesis, which was derived using a similar rationale as H1 and H2.

Furthermore, we researched students' attitude toward programming, understanding of programming basics, and understanding of programming concepts. Programming basics were investigated using actual problems. Programming concepts were evaluated using questionnaires and by beginners working on a problem. The same questionnaire was administered before (BQ) and after (AQ) the lecture to assess the change in programming attitude. Based on [11], we used five factors to assess attitude: Interest, Difficulty, Usefulness, Fun, and Willingness. Willingness is included because a desire to learn is an important element. We implemented the questionnaires on programming attitude before and after the lecture (Appendix. 1). Each question was evaluated using the six stages of the Likert scale (1: Strongly disagree, 2: Disagree, 3: Somewhat disagree, 4: Somewhat agree, 5: Agree and 6: Strongly agree). The Likert scale was set to six stages to eliminate an intermediate value, allowing the responses to be clearly divide into "can" and "cannot." For Q1, Q3, Q4, Q5, Q6, Q8, Q9, and Q10, a higher score in the after-questionnaire indicates an improvement, whereas for Q2 and Q7 a lower score in the after-questionnaire indicates an improvement. Q11 and Q12 were created to research the participants' understanding of programming concepts, and were answered descriptively. Q13, Q14, and Q15 inquired about students' impressions of the lecture. These questions were used to verify the whether the course is perceived as fun.

We recruited participants via a website. Participants were primary and junior high school students in Japan ranging in age from 8 to about 15 years old. In addition, the application allowed the participants to select the course type (visual or text). The subjects were 36 students who were learning to program for the first time. They were divided into two groups: Visual (G1) (25 people) and Text (G2) (11 people). To address the imbalance by in the future, each group should have roughly the same number of participants.

V. RESULTS AND ANALYSIS

A. *Results and analysis of the attitude toward programming*

We analyzed the questionnaire results (Appendix 2) using the Wilcoxon rank sum test (p-value < 0.05). This test is valid for a non-normal distribution or a small sample size. Further, the number of valid responses has 22(G1) and 11(G2). We evaluated the following to address RQ1: **(A1)** Analyze the change as the simple averaged value (Table3); **(A2)** Implement a Wilcoxon rank sum test in the BQ and AQ of each group; (**A3)** Implement a Wilcoxon rank sum test for the results of BQ in G1 and G2; (**A4**) Implement a Wilcoxon rank sum test for the results of AQ in G1 and G2; (**A5**) Implement a Wilcoxon rank sum test for the change in BQ and AQ. Figure 3 shows the detail of analysis.

Q1 probed the interest in programming. The difference in the G1 average is 0.32 [5.23 (before) and 5.55 (after)], whereas the difference in the G2 average is 0.28 [5.27 (before) and 5.55 (after)]. Neither group shows a significant increase in programming interest.

Q2 assessed the difficulty. The difference in the G1 is –0.09 [3.91 (before) and 3.82 (after)]. A decrease in the average indicates that programming is perceived to be more difficult after the lecture. On the other hand, the difference for G2 increases by 0.46 [4.55 (before) to 4.09 (after)]. The increased difficulty in G1 may be due to the increased work (e.g., drag-and-drop and more instructions) and the fact that programming occurs in 3D space. G2 shows a very slight improvement.

Q3 inquired about whether programming is useful. There is no change for G1 [5.68 (before) and 5.68 (after)]. G2 shows an improvement of 0.28 after the lecture [5.36 (before) and 5.64 (after)].

Q4 evaluated the fun of programming. Visual-based shows an average improvement of 0.41 the lecture [5.27 (before) and 5.68 (after)], whereas G2 only improves 0.19 [5.45 (before) and 5.64 (after)]. Thus, G2 is evaluated as more fun.

Q5 asked about motivation. After the G1 lecture, motivation increases by 0.27 [5.32 (before) and 5.59 (after)], whereas the value does not change (5.36) for G2, indicating that G1 increases the willingness to learn programming.

Q6 investigated whether turtle programming is interesting. After the lecture, the value for G1 increases by 0.82 [4.91 (before) and 5.73 (after)], while that for G2 increases by 0.27 [5.18 (before) and 5.45 (after)]. Although both improved, the change for G1 is more significant, indicating that it can be used to control the turtle.

Q7 examines the difficulty of Turtle programming. The value for G1 increases slightly (0.04) after the lecture, suggesting that programming is perceived as more difficult after the lecture. In contrast, the mean value for G2 is significantly reduced by -1.46, implying that Turtle programming is simple. However, this difference between the two methods may be because G2 involves less work or because fewer students participated.

Q8 inquired about the usefulness of turtle programming. G1 improves by 0.64 [5.00 (before) and 5.64 (after)]. On the other hand, the value for G2 decreases by 0.28 [5.55 (before) and 5.27 (after)], demonstrating that G1 is more convenient.

Q9 asked whether turtle programming is fun. Both methods are perceived as fun. G1 increases by 0.48 [5.27 (before) and 5.68 (after)], while G2 increases by 0.28 [5.36 (before) and 5.09 (after)].

Q10 assessed motivation for turtle programming. After the lecture, G1 improves motivation (0.77) [4.91 (before) and 5.68 (after)], but the willingness to learn decreases for text-based programming (–0.27) [5.36 (before) and 5.09 (after)]. These results show that G1 generally improves the attitude toward program, but that the G2 is perceived as easier to learn. Furthermore, G1 improves the turtle programming.

For a more detailed analysis, A2 used the Wilcoxon rank sum test to evaluate the before and after survey results for each group (Appendix 2). The G1 shows a significant difference in the after questionnaire for Q6 (interest in turtle programming) and Q10 (future learning of turtle programming), suggesting that the lecture enhances interest in the turtle program. However, the G2 responses are statistically insignificant, except for Q7 (turtle programming difficulty). The responses tend to significantly differ for the turtle program specific questions, and the G1 responses are more significant with respect to the attitude toward programming.

Next, A3 and A4 employed the Wilcoxon rank sum tests in the before-questionnaire (BQ) and the after-questionnaire (AQ) between the two groups, respectively. The results are statistically insignificant.

Finally, A5 used the Wilcoxon rank sum test to evaluate the amount of change between the before-questionnaire (BQ) and the after-questionnaire (AQ) (Appendix 2). The future learning (Q10) and difficulty (Q7) are statistically different. Based on the results from A1 after the lecture, the G2 felt that turtle programming is easier than the G1. However, the G1 is more willing to learn in the future.

Based on the all results, the G1 shows the most improvement and the results tend to be more significant than the G2. These observations confirm hypothesis H1 that visual-based programming is adequate for introductory programming learning for beginners. One limitation of our results is the population size. In the future, more experimental data should be accumulated.

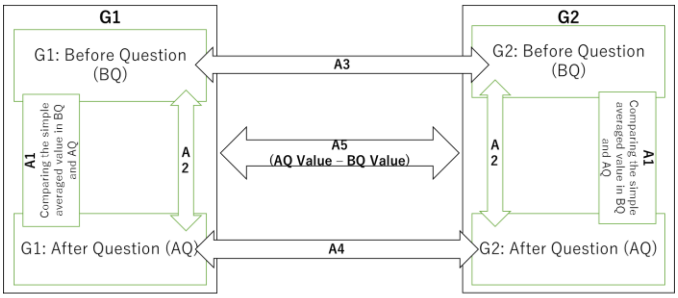


Fig. 3. Detail of analysis

TABLE III. CHANGE IN THE AVERAGE VALUES

	G1			G2		
Q	**Before**	**After**	**Evaluation**	**Before**	**After**	**Evaluation**
Q1	5.23	5.55	Improvement	5.27	5.55	Improvement
Q2	3.82	3.91	Degradation	4.55	4.09	Improvement
Q3	5.68	5.68	No change	5.36	5.64	Improvement
Q4	5.27	5.68	Improvement	5.45	5.64	Improvement
Q5	5.32	5.59	Improvement	5.36	5.36	No change
Q6	4.91	5.73	Improvement	5.18	5.45	Improvement
Q7	3.91	3.95	Degradation	4.55	3.09	Improvement
Q8	5.00	5.64	Improvement	5.55	5.27	Degradation
Q9	5.27	5.68	Improvement	5.36	5.64	Improvement
Q10	4.91	5.68	Improvement	5.36	5.09	Degradation

B. *Understanding Programming Basics*

To address RQ2, the lecture included six problems (P1~P6) that the learners answered. Because P6 could not be executed in the G2, the results of P6 were not considered. Each learner self-declared when a problem was complete and then took a screenshot to confirm the solution. Figure 4 shows the number of correct and incorrect (including blank) responses as well as the percentage of correct answers. Although the number of responses is small, the learners in the G1 have a better response rate and a higher overall accuracy rate than the G2). Both groups have similar percentages for P1 (move turtle), but the G2 has a lower motivation to complete P2 (add blocks) because it is too easy. On the other hand, the G2 learned how to move the turtle using text-programming, and successfully executed P3 (stack blocks) using the "for statement." The G1 excelled with P4 (create staircase) and P5 (avoid TNT blocks) because "if-statements" are easier to understand using visuals. Hence, visual-programming is superior to text-programming for understanding programming basics, confirming hypothesis H2. However, the low response rate remains an issue. Additional data and analysis are necessary to verify the superiority of visual-programming to text-programming.

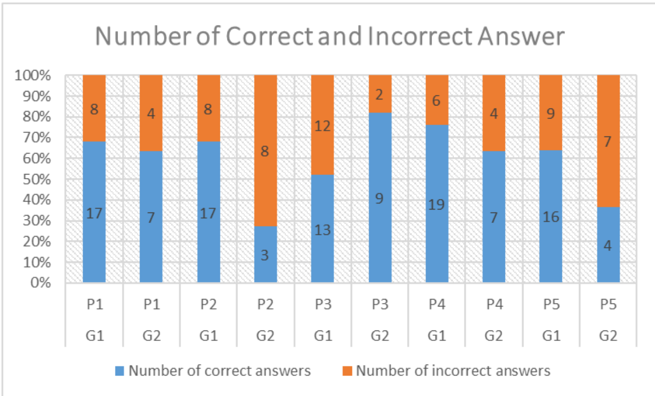


Fig. 4. Number of coreect and incorrect answer

C. *Understanding programming concepts*

To solve RQ3, we implemented a questionnaire and free problem in the lecture. The questionnaire contained the following two questions:

- What is loop handling?

G1: "To repeat a process."

G2: "A process is repeated the number of times specified."

- What is a conditional branch?

G1: "A designated treatment for the condition."

G2: "A program works when meeting some condition."

Both groups have similar comments. In addition, the free problem was to create the first letter of the learner's first name. Both groups utilized many iterations, indicating that a conditional branch is a difficult concept to grasp. The results between the groups are statistically insignificant, confirming that the abstraction level of the visual language is close to that of the text language.

D. *Impressions of the lecture*

Q13, Q14, and Q15 inquired about the impression of the lecture. In addition, the number of valid responses has 24(G1) and 11(G2). Figure 5, which shows the results of Q13 and Q14, indicates that overall the results are positive, but not significantly different. We asked for course feedback in Q13 (Was the course fun?) and Q14 (Do you want continue with this kind of learning?) Figure 4 shows the results. Q13 for G1 has a positive result; 20 people (95.2%) evaluated visual-based programming as "fun" or "a lot of fun." All 11 G2 participants indicated that it is fun. With regard to the free response (Q15), more than 10 people indicated that, "I want to do more." Overall, 90% gave a positive evaluation, indicating that programming education using a game can be fully utilized in an introductory course. Some of the written comments included: “I will use Minecraft with Turtle at home (G1)”. “Also, I want to learn in the lecture (G1)”. “This lecture is difficult (G2)”. “They taught me kindness (G2)”.

Although participants said that “the learning was fun and enjoy"(Q15), it is possible that they only enjoyed playing Minecraft and not participating in the lecture. Regardless, programming learning can be enjoyable when using Minecraft.

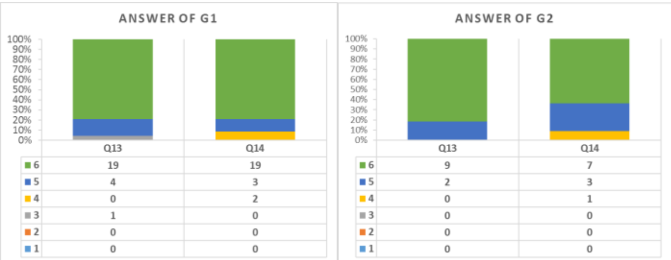


Fig. 5. Results of Q13 and Q14

VI. DISCUSSION

A. *Result of RQs and hypotheses*

RQ1 is “Do visual-based programming and text-based programming induce different attitudes toward programming?” The G1 shows a larger change, confirming H1. This result is attributed to that the fact that visual-based programming is more intuitive and it is easier to manipulate the turtle using visual-based programming.

RQ2 is “Does the understanding of programming basics vary between visual-based and text-based programming?” Although H2 indicates that visual-based programming is better due to the high percentage of correct answers, the low response rate questions the credibility of the results. Therefore, H2 cannot be confirmed using the results of this study.

RQ3 is “Does the understanding of programming concepts differ between visual-based and text-based programming?” There is no significant difference, which is inconsistent with H3. In addition, conditional branch is a difficult concept to grasp, and the abstraction level language is the same in both environments.

B. *Threats to validity*

We noted three issues with the results of the experiments.

First, learners in the G1 felt programming is more difficult after the lecture, which may be a consequence of the lecture design. In particular, visual-based programming involves many mouse operations such drag-and-drop. Redesigning the lecture should resolve this. Additionally, this increased perceived difficulty may also be due to Minecraft itself, which is a 3D environment. In Minecraft, users must visualize the environment in the horizontal, vertical, and depth directions simultaneously, which may be overwhelming to inexperienced users.

Second, the submission rates to the problems (P1~P6) were low due to the self-assessment. Although implementing a paper test may increase the response rate, it may not resolve this issue. We are currently considering ways to resolve this issue.

Last, learners could select the programming method. Because the participants were recruited via the Internet and could register for either the visual or text lecture, the number of participants differed between the two groups.

VII. RELATED WORKS

Some studies have compared programming environments using Minecraft for programming education. Thomas W. Price et al. compared of the text environment to the visual environment [10]. Their study investigated the effect of "attitudes towards computing", "perceived difficulty of programming", and "performance" on a "programming activity". They found that the visual environment affected the performance of the programming activities of students, but the other effects were insignificant. Our study investigated the attitudes to programming in detail

Christophert Zorn et al. used the Mod of CodeBlock for a student lecture course in 2013 [11]. Their research, which compared the learning effect of block programming to text programming, found that block-programming increases student interest. Furthermore, Brett Wilkinson et al. executed a Workshop using ComputerCraft in 2013 [12]. We investigated the effects of the programming environment on learning by implementing two programming inputs.

VIII. CONCLUSION AND FUTURE WORK

Visual-based programming has positive significant results with respect to interest and motivation to learn turtle programming. In addition, it tends to produce higher accuracy rates than text-based programming. However, students felt visual-based programming is more difficult after the course. Additionally, both methods yield similar results with respect to programming concepts. This research has some limitations. First, learners in the G1 felt that programming is more difficult after the lecture. Second, the response rate is low, which is likely due to the self-assessment. Third, using a digital game as a learning tool leads to challenges such as students playing the game instead of learning the lesson. In the future, we plan to collect and analyze additional data as well as address the issue of selecting either the visual or text lecture.

REFERENCE

[1] Scratch, https://scratch.mit.edu/, [Online available]: Dec 2015

[2] Alice, http://www.alice.org/index.php, Online available]: Jul 2016

[3] Carlisle, Martin C., et al. "RAPTOR: a visual programming environment for teaching algorithmic problem solving." ACM SIGCSE Bulletin. Vol. 37. No. 1. ACM, 2005.

[4] Chiu, Chiung-Fang. "Introducing Scratch as the Fundamental to Study App Inventor Programming." Learning and Teaching in Computing and Engineering (LaTiCE), 2015 International Conference on. IEEE, 2015.

[5] Kölling, Michael, Neil CC Brown, and Amjad Altadmri. "Frame-Based Editing: Easing the Transition from Blocks to Text-Based

Programming." Proceedings of the Workshop in Primary and Secondary Computing Education on. ACM, 2015.

[6] Matsuzawa, Yoshiaki, et al. "Language Migration in non-CS Introductory Programming through Mutual Language Translation Environment." Proceedings of the 46th ACM Technical Symposium on Computer Science Education. ACM, 2015.

[7] Mojang. Minecraft. https://minecraft.net/. [Online available]: Aug 2015.

[8] ComputerCraft, http://www.computercraft.info/, [Online available]: Aug 2015.

[9] ComputerCraftEdu, http://computercraftedu.com/, TeacherGaming LCC, [Online available]: Aug 2015.

[10] Price, Thomas W., and Tiffany Barnes. "Comparing Textual and Block Interfaces in a Novice Programming Environment." Proceedings of the eleventh annual International Conference on International Computing Education Research. ACM, 2015.

[11] Zorn, C., Wingrave, C. A., Charbonneau, E., & LaViola Jr, J. J, “Exploring Minecraft as a conduit for increasing interest in programming”, The 8th International Conference on the Foundations of Digital Games, pp. 352-359, 2013

[12] Wilkinson, B., Williams, N., & Armstrong, P. Improving Student Understanding, Application and Syn-thesis of Computer Programming Concepts with Minecraft.

[13] Saito, Daisuke, and Tsuneo Yamaura. "Applying the top-down approach to beginners in programming language education." Interactive Collaborative Learning (ICL), 2014 International Conference on. IEEE, 2014.

APPENDIX

TABLE IV. (APPENDIX 1) QUESTIONNAIRES

Question No.	Question	Impression
	Programming attitude questions	
Q1	Are you interested in programming?	Interest
Q2	Do you think that learning to program is difficult?	Difficulty
Q3	Do you think that knowing how to program is useful?	Useful
Q4	Do you think programming is fun?	Fun
Q5	Do you want to learn to program?	Willingness
Q6	Are you interested in the turtle program?	Interest
Q7	Do you think that the learning the turtle program is difficult?	Difficulty
Q8	Do you think that knowing the turtle program is useful?	Useful
Q9	Do you think turtle programming is fun?	Fun
Q10	Do you want to learn turtle programming?	Willingness
	Programming concept questions	
Q11	What is a conditional branch?	Conditional branching
Q12	What is loop handling?	Repeat
	Impressions of the lecture questions	
Q13	Did you enjoy learning to program using Minecraft?	N?A
Q14	Did the lecture make you want to learn?	
Q15	Free description	

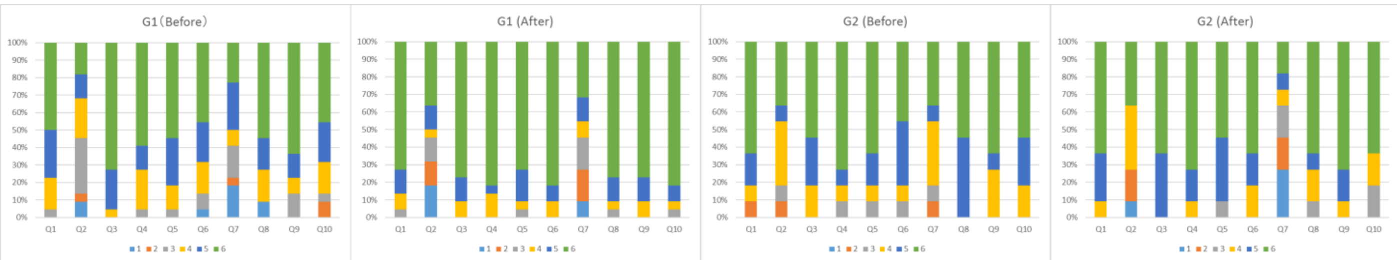


Fig. 6. (APPENDIX 2) Result of Questionnaires

TABLE V. (APPENDIX 3) RESULTS OF THE WILCOXON RANK-SUM TEST (95% CONFIDENCE INTERVALS(P-VALUE < 0.05)

Q	A2 (G1), z-score	A2 (G1), p-value	A2 (G2), z-score	A2 (G2), p-value	A5, z-score	A5, p-value
Q1	-1.23	0.22	-0.16	0.87	-0.55	0.58
Q2	-0.26	0.80	0.49	0.62	-1.07	0.29
Q3	-0.18	0.87	-0.62	0.53	0.73	0.47
Q4	-1.29	0.20	-0.13	0.9	-0.31	0.76
Q5	-1.06	0.29	0.2	0.84	-0.8	0.42
Q6	**-2.18**	**0.03***	-0.66	0.51	-0.94	0.35
Q7	-0.15	0.89	1.84	0.07	**-2.06**	**0.04***
Q8	-1.41	0.16	0.13	0.9	-1.49	0.14
Q9	-0.95	0.34	-0.53	0.6	-0.13	0.89
Q10	**-2.12**	**0.03***	0.16	0.87	**-2.08**	**0.04***

*** : p < 0.05); There is a significant difference**