Journal of Information Technology Education: Research

An Official Publication of the Informing Science Institute
InformingScience.org

JITEResearch.org

Volume 16, 2017

# Comparison of Text-Based and Visual-Based Programming Input Methods for First-time Learners

| | | |
|---|---|---|
| Daisuke Saito * | Graduate School of Fundamental Science and Engineering, Waseda University, Tokyo, Japan | d.saito@fuji.waseda.jp |
| Hironori Washizaki | Global Software Engineering Laboratory, Waseda University, Tokyo, Japan | washizaki@waseda.jp |
| Yoshiaki Fukazawa | Department of Information and Computer Science, Waseda University, Tokyo, Japan | fukazawa@waseda.jp |

* Corresponding author

## Abstract

| | |
|---|---|
| Aim/Purpose | When learning to program, both text-based and visual-based input methods are common. However, it is unclear which method is more appropriate for first-time learners (first learners). |
| Background | The differences in the learning effect between text-based and visual-based input methods for first learners are compared the using a questionnaire and problems to assess first learners' understanding of programming. In addition, we study the benefits and feasibility of both methods. |
| Methodology | In this research, we used the sandbox game Minecraft and the extended function ComputerCraftEdu (CCEdu). CCEdu provides a Lua programming environments for the two (text and visual) methods inside Minecraft. We conducted a lecture course on both methods for first learners in Japan ranging in age from 6 to about 15 years old. The lecture taught the basics and concepts of programming. Furthermore, we implemented a questionnaire about the attitude of programming before and after the lecture. |
| Contribution | This research is more than a comparison between the visual method and the text method. It compares visual input and text input methods in the same environment. It clearly shows the difference between the programming learning effects of visual input and text input for first learners. In addition, it shows the more suitable input method for introductory education of first learners in |

Accepted by Editor Tian Luo │Received: February 26 2017│ Revised: May 5, May 25, June 11, 2017 │ Accepted: June 19, 2017.

Cite as: Saito, D., Washizaki, H., & Fukazawa, Y. (2017). Comparison of text-based and visual-based programming input methods for first-time learners. *Journal of Information Technology Education: Research, 16,* 209-226. Retrieved from http://www.informingscience.org/Publications/3775

| | |
|---|---|
| | programming learning. |
| Findings | The following results are revealed: (1) The visual input method induces a larger change in attitude toward programming; (2) The number of operations and input quantity influence both groups; (3) The overall results suggest that a visual input is advantageous in a programming implementation environment for first learners. |
| Impact on Society | A visual input method is better suited for first learners as it improves the attitude toward programming. |
| Future Research | In the future, we plan to collect and analyze additional data as well as elucidate the correlation between attitudes and understanding of programming. |
| Keywords | programming-learning, minecraft. programming input method, game-based learning |

# INTRODUCTION

Both visual input and text input methods have been used to teach programming to beginners. In programming, first-time learners (first learners) tend to employ visual input methods instead of text methods. Although some studies have found that visual inputs are superior, others have shown that text inputs can be adapted to first learners (Saito & Yamaura, 2013). In addition, several tools have been developed for both types of programming in the same environment (Bau, Bau, Dawson, & Pickens, 2015; Fraser, 2013). However, few studies have compared the learning effects by input method. Hence, whether one programming method (text or visual) is more suitable for introductory education remains unknown (Price & Barnes, 2015; Weintrop & Holbert, 2017).

Herein the differences between visual and text input methods are investigated in the same Lua programming environment to determine if the input method influences the learning effect. Specifically, this research examines the following Research Questions (RQs):

- **Research Question 1 (RQ1)**: Does a visual-based input method induce a different attitude toward programming than a text-based input method?
- **Research Question2 (RQ2)**: Does the understanding of programming between visual-based and text-based input methods differ?

RQ1 assesses whether the programming method is suitable for an introductory environment. This RQ can elucidate the attitude of first learners toward programming by input method, which should assist in selecting the most suitable method for introductory programming. RQ2 evaluates the understanding of programming basics. Furthermore, it examines the understanding of programming concepts by focusing on a sequential execution, conditional branching, and repetition. This RQ can reveal which method is more suitable for learning. These RQs should help determine the more 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 first learners' motivation to learn.

The rest of this paper organized as follows. The Background Section explains the background and previous works. The Lecture Design Section discusses the lecture design used to compare the two methods. The Experiment Section and the Analysis Method Section describe the experiment using the lecture design and the analytical techniques, respectively. The Results Section overviews the results. The Discussion Section evaluates the results. Finally, the Conclusion and Limitations Section provides the conclusions, threats to validity, and future work.

## BACKGROUND

### *PROGRAMMING LEARNING FOR FIRST LEARNERS*

It is often noted that beginners have difficulty learning to program (Gross & Powers, 2005; Kelleher & Pausch, 2005). Several studies have been conducted to address this issue. Some studies used a visual method like Scratch developed by MIT (Chiu, 2015; Maloney, Resnick, Rusk, Silverman, & Eastmond, 2010; Sáez-López, Román-González, & Vázquez-Cano, 2016). A different study used a text method such as C (Saito & Yamaura, 2013). Other studies used both visual and text methods for Project-based Learning for programming based on problem-solving (O'Kelly & Gibson, 2006) and Game-based Learning (Jiau, Chen, & Ssu, 2009; Long, 2007; Vasilateanu, Wyrazic, & Pavaloiu, 2016). In addition, some works investigated attitudes toward programming (Du, Wimmer, & Rada, 2016).

Each method has its own learning effect. Some success has been reported for first learners using these methods. However, which programming input method (visual or text) is more suitable for first learners remains unknown. In addition, the learning effect for each method is unclear. Based on this situation, this paper focuses on the input method and compares the learning effect in the same programming environment. It is intended to serve as a reference for educators when selecting an input method to teach first learners.

### *TWO INPUT METHODS*

In this paper, we evaluate the learning effect of text inputs and visual inputs in the same programming language. A comparison of the learning effects in text and visual methods can be traced back to the Dual-coding theory (DCT) raised by Paivio (Clark & Paivio, 1991; Paivio, 2013). In this theory, human information processing can be divided into two systems: language and non-language. Language systems are character information such as characters and voices, while non-language systems are sensory information such as images. These affect human recognition (Clark & Paivio, 1991).

Studies have examined characters and images using DCT. One study investigated the influence of student's prior knowledge on a computer-based physical lesson learning due to differences in the presentation format (text, images, animation) (ChanLin, 2001). This study found that using images for beginners is useful for descriptive learning and procedural learning. Another study concluded that it is more effective to use images and letters together (Mayer, 2003). Furthermore, Eitel and Scheiter (2015) reviewed 42 studies on the presentation order of texts and images while learning. The boundary condition to determine whether it is better to use the first process as an image or text is stated as the relative complexity of the image and the text. Unlike this study, which focuses on programming languages, these studies focused on multimedia learning.

A programming language can be expressed as text representations or visual representations. For example, visual programming languages like Scratch (Maloney et al., 2010) or Alice (Dann, Cooper, & Pausch, 2011) use drag and drop of visual inputs for program learning. A visual language is suitable as a language initially performed by first learners who are unfamiliar with a programming language. Furthermore, text programming languages such as Python and JavaScript use a keyboard to type inputs. Text languages can be more sophisticated than visual languages. Although a text language is better suited if the purpose is clear, learners must possess sufficient typing skills. In addition, some researchers have investigated the transition to text-based programming from visual-based programming (Bau et al., 2015; Kölling, Brown, & Altadmri, 2015; Lahtinen, Ala-Mutka, & Järvinen, 2005). Hence, the research results implemented in the field of multimedia are applicable for first learners in programming.

In a study comparing programming methods, visual programming methods are noted to be an easy educational method for educators (Weintrop & Wilensky, 2015a). Studies on programming in higher education indicate that visual-based languages produce better results than alternative programs (Weintrop & Wilensky, 2015b). One study developed an extended function of Codeblock, which expands the visual programming function to Minecraft. It reported that this tool results in improved recognition of programming. Although this study compared the visual program function to text environments, a significant difference was not detected (Zorn, Wingrave, Charbonneau, & LaViola, 2013).

Several reports in multimedia and programming learning have indicated that the visual method is suitable for first learners. In other words, they suggest that using a visual input method may be more advantageous for first learners. However, programming involves both visual information as well as behavioral aspects such as input of programs and confirmation of execution results. It is difficult to support all results in the multimedia field. In addition, there is no significant difference in recognition of programming compared to the text environment (Zorn et al., 2013). Consequently, the proper input method for first learners is not definitive. This study uses visual inputs and text inputs at the same abstraction level built in the same environment to provide clear answers. Hence, the comparison is based only on the input differences without the effect of the environment. Additionally, this study strives to include younger participants. Table 1 highlights the difference from recent studies.

**Table 1. Differences between this study and recent research**

| | **This research** | **(Zorn et al., 2013)** | **(Price, & Barnes, 2015)** |
|---|---|---|---|
| **Visual input method** | CCEdu (Visual) | CodeBlock 1* (3D Block) | Tiled Grace 2* (Block) |
| **Text input method** | CCEdu (Text) | CodeBlock (Text) | Tiled Grace (Text) |
| **Representation of programming language** | Text and Picture | Text and Picture | Text |
| **Programming environment** | Minecraft | Minecraft(Visual) and Web(text) | Web |
| **Correspondence between Visual and Text** | 1 on 1 | 1 on 1 | 1 on 1 |
| **Target age** | 6-15 | 18 -51 | 11(Block), 12 (Text) 3* |

***1**: Codeblock is an additional feature that provides a programming environment on Minecraft. It has one environment for programming with 3D Block and another for programming with text.
***2**: Tile Grace is a web environment for Grace in an educational programming language. This environment has a programming environment for text and a block programming environment expressed in the text.
***3**: It expressed as 6th grade and 7th grade (K-12).

### *MINECRAFT AND COMPUTERCRAFTEDU WITH PROGRAMMING LEARNING*

We used Minecraft for program learning. Minecraft is an internationally popular sandbox game, which involves using various materials to build objects and structures. Minecraft as an educational tool has been demonstrated in mathematics and science (Bayliss, 2012; Gallagher, 2014).

ComputerCraft is a Mod that adds the function of Lua in a programming language. Previous research employed a workshop to study programming language education using ComputerCraft based on the revised taxonomy of Bloom (Wilkinson, Williams, & Armstrong, 2013). Student motivation improved using ComputeCraft. In addition, it has been reported that programming language education using ComputerCraft is beneficial (Wilkinson et al., 2013).

We used ComputerCraftEdu (CCEdu), which is the education version of ComputerCraft. This mod has two environments for programming: text-based and visual-based (Figure 1). Text-based programming can be controlled in Minecraft using the same method as general text programming, while visual-based programming employs illustration 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.


**Figure 1. Two programming environments in CCEdu**

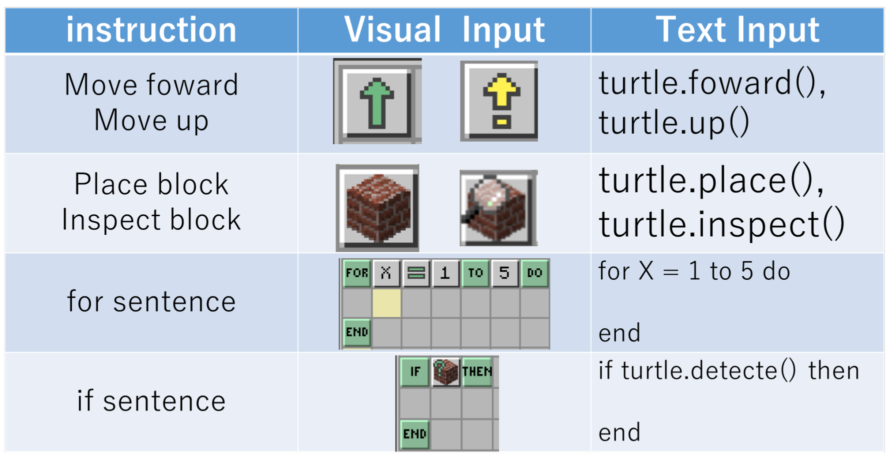



**Figure 2. Relationship between the two input methods**

## LECTURE DESIGN

We implemented two types of lectures (visual-based and text-based). Each lecture, which was designed for elementary and junior high school students in Japan, was configured as a short course. Both covered the same contents. Each consisted of a tutorial, sequential execution, repeat, conditional branching, and a free problem. The order of the lecture contents was as follows. (1) The tutorial content focused on operations in Minecraft and ComputerCraftEdu. (2) Sequential execution involved learning a sequential execution, which is a programming basic. The example in the lecture was to move a turtle and place a block in Minecraft. We taught turtle instructions to move forward, back, left, right, up, or down. (3) Loop 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. (4) Conditional was explained using the "if sentence" to avoid a block. The lecture used two examples: "avoid obstacles" and "remove TNT". (5) Finally, a free problem was used to assess the students' programming skills. The free problem was to create one alphabet character. In addition, to gauge the understanding of programming, the lecture included six problems (Table 2).

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

**Table 2. Detail of problem contents**

| *#* | *Problem Contents* | *Survey Category* |
|---|---|---|
| *P1* | Move the turtle three steps, rotate left, and move two more steps. | Sequential |
| *P2* | Add four blocks. | Sequential |
| *P3* | Stack eight blocks. | Loop |
| *P4* | Create a stairway with eight steps. | Loop |
| *P5* | If a TNT block is in front of the turtle, avoid it. | Conditional |
| *P6* | If a diamond block is in front of the turtle, mine it. | Conditional |

# EXPERIMENTS

We confirmed whether the text or visual method is more suitable for an introductory education by comparative experiments based on the “Lecture Design” described in the previous section. In addition, we developed two hypotheses that correspond to the RQs:

- **Hypotheses 1(H1)**: The visual input programming lecture induces a larger change in attitude toward programming.
- **Hypotheses 2(H2)**: Programming is easier to understand using the visual input method.

H1 is the RQ1 hypothesis. It speculated that the change in attitude toward programming is more significant for the visual input group because the visual input method is more intuitive than the text input method. H2 is the RQ2 hypothesis. Similar to the rationale for H1, it should be easier to comprehend programming using visual inputs.

## *PARTICIPANTS*

We recruited participants via a website in 2015 and 2016. Participants were primary and junior high school students in Japan ranging in age from 6 to about 15 years old. The application allowed participants to select the course type (visual or text). A total of 72 subjects participated. In each year, 36 students responded to the recruitment targeting first learners. Based on the participant’s preference, they were divided into the Visual Group (VG) and the Text Group (TG). Learners attended the lecture by the group. VG had 46 participants (2015: 25 people, 2016: 21 people), while TG had 26 (2015: 11 people, 2016: 15 people).

## *QUESTIONNAIRE*

The same questionnaire was administered twice to assess the change in programming attitude: Before Questionnaire (BQ: Q1B - Q10B) and After Questionnaire (AQ: Q1A - Q10A) (Table 3). Based on Zorn et al. (2013), we used five factors to assess attitude: Interest, Difficulty, Usefulness, Fun, and Willingness. Willingness is included because the desire to learn is an important element. 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 divided into “can” and “cannot”. For all questions except Q2 and Q7, a higher score in the after-questionnaire indicated an improvement. For Q2 and Q7, a lower score in the after-questionnaire indicated an improvement. Furthermore, we created two questions (Q11, Q12) to assess the participants' understanding of programming.

**Table 3. Questionnaire about attitude toward programming and understanding programming**

| # | Attitude Toward Programming Question | Survey Category |
|---|---|---|
| **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? | Usefulness |
| **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(Turtle) |
| **Q7** | Do you think that the learning the turtle program is difficult? | Difficulty(Turtle) |
| **Q8** | Do you think that knowing how to turtle program is useful? | Usefulness(Turtle) |
| **Q9** | Do you think turtle programming is fun? | Fun(Turtle) |
| **Q10** | Do you want to learn to turtle program? | Willingness(Turtle) |
| | Understanding Programming questions | |
| **Q11** | What is a conditional? | Conditional |
| **Q12** | What is a loop? | Loop |

## ANALYSIS METHOD

We tested the normality of the results of each questionnaire to determine the appropriate analysis method using the Shapiro-Wilk test. This test evaluates the normality of a given population. A p-value ≤ 0.05 indicates normality is not recognized. We were unable to confirm that the data follows a normal distribution in all populations (Table 4). Hence, the Wilcoxon signed-rank test and the Wilcoxon rank sum test were adopted. Unlike t-tests, these tests can be used without normality. In addition, the Wilcoxon signed-rank test is a significant difference test between two groups with correspondence. The Wilcoxon rank sum test is a significant difference test between two groups without correspondence.

**Table 4. Shapiro-Wilk test results**

| | **VG** | | | | **TG** | | | |
|---|---|---|---|---|---|---|---|---|
| | **BQ** | | **AQ** | | **BQ** | | **AQ** | |
| | W | p | W | p | W | p | W | p |
| **Q1** | 0.841149 | **8.11E-05** | 0.702422 | **1.80E-07** | 0.699444 | **5.21E-06** | 0.593814 | **2.51E-07** |
| **Q2** | 0.924076 | **0.01311** | 0.882925 | **0.00087** | 0.889436 | **0.009165** | 0.884148 | **0.007052** |
| **Q3** | 0.648226 | **2.67E-08** | 0.624026 | **1.21E-08** | 0.668489 | **2.02E-06** | 0.54512 | **7.30E-08** |
| **Q4** | 0.757055 | **1.56E-06** | 0.661249 | **4.15E-08** | 0.642564 | **9.52E-07** | 0.527394 | **4.76E-08** |
| **Q5** | 0.776291 | **3.56E-06** | 0.664308 | **4.61E-08** | 0.634614 | **7.60E-07** | 0.690753 | **3.97E-06** |
| **Q6** | 0.804084 | **1.27E-05** | 0.664948 | **4.71E-08** | 0.760626 | **4.04E-05** | 0.66596 | **1.88E-06** |
| **Q7** | 0.904273 | **0.003372** | 0.911177 | **0.00535** | 0.861033 | **0.002346** | 0.869628 | **0.003503** |
| **Q8** | 0.796619 | **8.93E-06** | 0.673467 | **6.34E-08** | 0.696393 | **4.73E-06** | 0.781564 | **8.69E-05** |
| **Q9** | 0.769849 | **2.69E-06** | 0.690709 | **1.17E-07** | 0.708225 | **6.88E-06** | 0.579284 | **1.72E-07** |
| **Q10** | 0.814639 | **2.11E-05** | 0.6989 | **1.58E-07** | 0.675584 | **2.50E-06** | 0.70287 | **5.81E-06** |

The number of valid responses was 38 (VG) and 26 (TG). We evaluated the following to address RQ1: (A1) Analyze the change in a simple averaged value; (A2) Implement a Wilcoxon signed-rank test in the BQ and AQ by group; (A3) Implement a Wilcoxon rank sum test for the results of BQ by group; (A4) Implement a Wilcoxon rank sum test for the results of AQ by group; (A5) Implement a Wilcoxon rank sum test for the change in BQ and AQ. Figure 3 shows the details of the analysis.

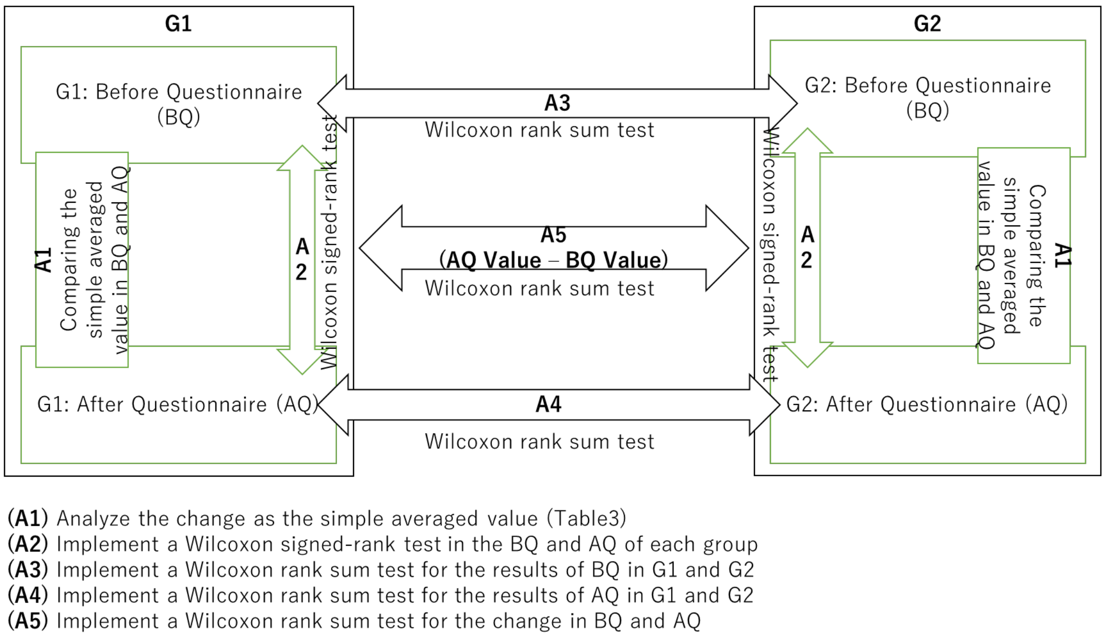


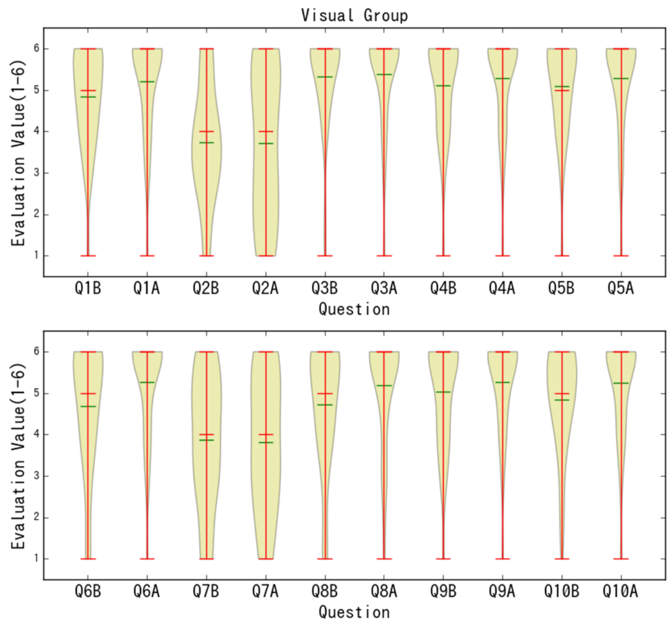

**Figure 3 Analysis details**

# RESULT

## *ATTITUDE TOWARD PROGRAMMING*

### Questionnaire result

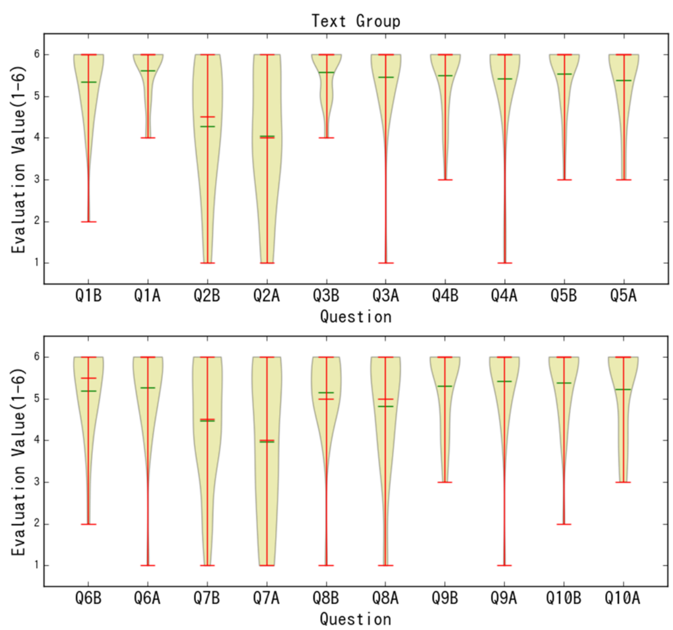
Figure 4 (VG) and Figure 5 (TG) show the results of the questionnaires (Q1-Q10) using violin plots. A violin plot expresses the distribution of data, allowing the distribution density, average value, and median value to be confirmed. It is possible to verify the change in the value of the Likert scale before and after the lecture and the distribution density. The green lines in the plot (Figures 4 and 5) show the average values. After the lecture, the results for most categories improved for VG. On the other hand, the results of TG decreased to an overall negative attitude, except for interest (Q1, Q6) and difficulty (Q2, Q7) in programming, which showed an improvement. VG had a larger change in values than TG. Hence, visual inputs may be more suitable for first learners than text inputs. However, TG had a larger improvement for the difficulty of programming than VG. In addition, there was no difference between BQ (before questionnaire) and AQ (after questionnaire) because the answers regarding difficulty in VG had many positive attitudes in the BQ.




| Questions ID | Survey Category |
|---|---|
| Q1(Q1A, Q1B) | Interest |
| Q2(Q2A, Q2B) | Difficulty |
| Q3(Q3A, Q3B) | Usefulness |
| Q4(Q4A, Q4B) | Fun |
| Q5(Q5A, Q5B) | Willingness |
| Q6(Q6A, Q6B) | Interest(Turtle) |
| Q7(Q7A, Q7B) | Difficulty(Turtle) |
| Q8(Q8A, Q8B) | Usefulness(Turtle) |
| Q9(Q9A, Q9B) | Fun(Turtle) |
| Q10(Q10A, Q10B) | Willingness(Turtle) |

| | Average value | | | | Standard deviation | | | |
|---|---|---|---|---|---|---|---|---|
| | VG | | TG | | VG | | TG | |
| | Before | After | Before | After | Before | After | Before | After |
| Q1 | 4.842 | 5.211 | 5.346 | 5.615 | 1.182 | 1.196 | 0.958 | 0.684 |
| Q2 | 3.737 | 3.711 | 4.269 | 4.038 | 1.408 | 1.805 | 1.533 | 1.675 |
| Q3 | 5.316 | 5.368 | 5.577 | 5.462 | 1.126 | 1.134 | 0.631 | 1.046 |
| Q4 | 5.105 | 5.289 | 5.5 | 5.423 | 1.165 | 1.168 | 0.844 | 1.246 |
| Q5 | 5.079 | 5.289 | 5.538 | 5.385 | 1.133 | 1.168 | 0.796 | 0.923 |
| Q6 | 4.684 | 5.263 | 5.192 | 5.269 | 1.524 | 1.207 | 1.038 | 1.094 |
| Q7 | 3.868 | 3.816 | 4.462 | 3.962 | 1.609 | 1.636 | 1.474 | 1.808 |
| Q8 | 4.711 | 5.184 | 5.154 | 4.808 | 1.503 | 1.315 | 1.063 | 1.468 |
| Q9 | 5.026 | 5.263 | 5.308 | 5.423 | 1.246 | 1.14 | 0.991 | 1.08 |
| Q10 | 4.842 | 5.237 | 5.385 | 5.231 | 1.268 | 1.157 | 0.964 | 1.085 |

**Figure 4. Visual group results**




| Questions ID | Survey Category |
|---|---|
| Q1(Q1A, Q1B) | Interest |
| Q2(Q2A, Q2B) | Difficulty |
| Q3(Q3A, Q3B) | Usefulness |
| Q4(Q4A, Q4B) | Fun |
| Q5(Q5A, Q5B) | Willingness |
| Q6(Q6A, Q6B) | Interest(Turtle) |
| Q7(Q7A, Q7B) | Difficulty(Turtle) |
| Q8(Q8A, Q8B) | Usefulness(Turtle) |
| Q9(Q9A, Q9B) | Fun(Turtle) |
| Q10(Q10A, Q10B) | Willingness(Turtle) |

| | Average value | | | | Standard deviation | | | |
|---|---|---|---|---|---|---|---|---|
| | VG | | TG | | VG | | TG | |
| | Before | After | Before | After | Before | After | Before | After |
| Q1 | 4.842 | 5.211 | 5.346 | 5.615 | 1.182 | 1.196 | 0.958 | 0.684 |
| Q2 | 3.737 | 3.711 | 4.269 | 4.038 | 1.408 | 1.805 | 1.533 | 1.675 |
| Q3 | 5.316 | 5.368 | 5.577 | 5.462 | 1.126 | 1.134 | 0.631 | 1.046 |
| Q4 | 5.105 | 5.289 | 5.5 | 5.423 | 1.165 | 1.168 | 0.844 | 1.246 |
| Q5 | 5.079 | 5.289 | 5.538 | 5.385 | 1.133 | 1.168 | 0.796 | 0.923 |
| Q6 | 4.684 | 5.263 | 5.192 | 5.269 | 1.524 | 1.207 | 1.038 | 1.094 |
| Q7 | 3.868 | 3.816 | 4.462 | 3.962 | 1.609 | 1.636 | 1.474 | 1.808 |
| Q8 | 4.711 | 5.184 | 5.154 | 4.808 | 1.503 | 1.315 | 1.063 | 1.468 |
| Q9 | 5.026 | 5.263 | 5.308 | 5.423 | 1.246 | 1.14 | 0.991 | 1.08 |
| Q10 | 4.842 | 5.237 | 5.385 | 5.231 | 1.268 | 1.157 | 0.964 | 1.085 |

**Figure 5. Text group results**

## Analysis of the results

We analyzed the questionnaire results using the Wilcoxon signed-rank test and the Wilcoxon rank sum test (p-value < 0.05). Table 5 shows the results for A1, while Table 6 shows the results for A2 – A3. VG improved in all categories according to A1. In particular, the attitude towards turtle programming improved and the interest in turtle programming improved by about 0.6 points. However, some of the learners showed a lower value for attitude after the lecture. It is possible that some learners became bored with programming or were more absorbed in playing the game than programming. Some categories for TG also improved to a positive attitude, while others decreased to a negative attitude. The large amount of input necessary to program may induce a negative attitude.

**Table 5. Change in the average values**

| # | VG Before | VG After | VG CV | VG Evaluation | TG Before | TG After | TG CV | TG Evaluation |
|---|---|---|---|---|---|---|---|---|
| **Q1** | 4.842 | 5.211 | 0.368 | Improvement | 5.346 | 5.615 | 0.269 | Improvement |
| **Q2** | 3.737 | 3.711 | -0.026 | Improvement | 4.269 | 4.038 | -0.231 | Improvement |
| **Q3** | 5.316 | 5.368 | 0.053 | Improvement | 5.577 | 5.462 | -0.115 | Degradation |
| **Q4** | 5.105 | 5.289 | 0.184 | Improvement | 5.500 | 5.423 | -0.077 | Degradation |
| **Q5** | 5.079 | 5.289 | 0.211 | Improvement | 5.538 | 5.385 | -0.154 | Degradation |
| **Q6** | 4.684 | 5.263 | 0.579 | Improvement | 5.192 | 5.269 | 0.077 | Improvement |
| **Q7** | 3.868 | 3.816 | -0.053 | Improvement | 4.462 | 3.962 | -0.500 | Improvement |
| **Q8** | 4.711 | 5.184 | 0.474 | Improvement | 5.154 | 4.808 | -0.346 | Degradation |
| **Q9** | 5.026 | 5.263 | 0.237 | Improvement | 5.308 | 5.423 | 0.115 | Improvement |
| **Q10** | 4.842 | 5.237 | 0.395 | Improvement | 5.385 | 5.231 | -0.154 | Degradation |

**CV = Change Value (After – Before)**

**Table 6. Results of the Wilcoxon single rank test and the Wilcoxon Rank-Sum Test [95% Confidence Intervals (p-value < 0.05)]**

| | A2(VG) | | A2(TG) | | A3 | | A4 | | A5 | |
|---|---|---|---|---|---|---|---|---|---|---|
| # | **S** | **p** | **S** | **p** | **S** | **p** | **S** | **p** | **S** | **p** |
| Q1 | 32 | **0.029**** | 5.5 | 0.143 | -1.76 | **0.079*** | -1.05 | 0.293 | 0.40 | 0.687 |
| Q2 | 157.5 | 0.891 | 27 | 0.338 | -1.54 | 0.124 | -0.71 | 0.477 | 0.96 | 0.339 |
| Q3 | 53 | 0.672 | 13 | 0.863 | -0.52 | 0.603 | -0.04 | 0.967 | 0.03 | 0.978 |
| Q4 | 23 | 0.2 | 18 | 1 | -1.24 | 0.214 | -0.57 | 0.566 | 0.05 | 0.956 |
| Q5 | 22.5 | 0.183 | 20 | 0.427 | -1.61 | 0.107 | -0.01 | 0.989 | 1.00 | 0.315 |
| Q6 | 30.5 | **0.008**** | 35 | 0.439 | -1.1 | 0.271 | 0.4 | 0.692 | 1.09 | 0.274 |
| Q7 | 144.5 | 0.873 | 62 | 0.178 | -1.46 | 0.145 | -0.41 | 0.682 | 0.51 | 0.613 |
| Q8 | 35 | **0.045**** | 27.5 | 0.185 | -0.9 | 0.371 | 1.13 | 0.257 | 1.82 | **0.069*** |
| Q9 | 42 | 0.151 | 42 | 0.5 | -0.74 | 0.46 | -0.51 | 0.613 | 0.1 | 0.924 |
| Q10 | 34 | **0.070*** | 42 | 0.5 | -1.82 | **0.069*** | 0.05 | 0.956 | 1.35 | 0.176 |

**S = Statistics**
**p = p-value**
****(p<0.05): There is a significant difference**
***(0.05<=p<0.10): There is a significant trend**

The attitudes for interest and difficulty of programming became positive. Furthermore, the attitudes for interest, difficulty, and fun of turtle programming improved.

The results of A2 indicated a statistically significant difference in VG in the after questionnaire for Q1 (interest in programming, p = 0.029), Q6 (interest in turtle programming, p = 0.008), and Q8 (Useful of turtle programming, p = 0.045), suggesting that the lecture enhanced interest in programming. On the other hand, there was no significant difference in TG. The responses tended to significantly differ from the turtle programming specific questions for both groups. The VG responses were more significant on the attitude toward programming than TG.

Next, A3 involved a Wilcoxon rank sum test in the before questionnaire between the two groups. There were no significant differences, but marginal differences were observed for interest to programming ($p = 0.079$) and willingness for turtle programming ($p = 0.069$). The marginal differences are attributed to the negative values in the before questionnaire in VG.

The A4 analysis was the same as the A3 analysis, except the after questionnaires compared results. The results were statistically insignificant.

The A5 analysis was also carried out on the change in value. The usefulness of turtle program was marginally significant ($p = 0.069$).

Overall, VG had a larger positive change in the attitude toward programming than TG. However, both VG and TG showed a positive increase in interest in programming.

After the lecture, both groups responded that programming is difficult. TG showed a very slight improvement compared to VG [TG (A1: -0.231) vs. VG (A1: -0.026)], but the difference was insignificant. However, the results imply that the text method has a larger effect on reducing the difficulty level for programming than the visual input method.

Regarding the usefulness of programming, VG showed an improvement, while TG did not. However, the results did not significantly differ. The text input had more input responses than the visual input, which may be a factor in the decrease in TG.

Regarding the fun for programming, VG slightly increased, while TG slightly decreased. However, the analysis revealed that the difference was not significant. Similar to above, text input had more input responses than visual input, which may be a factor for the decrease in TG.

As for the willingness for programming, VG improved, while TG did not. However, the difference was not significant. The decline in the willingness in TG may be attributed to the decline in the fun for programming.

Both VG and TG showed an increased interest in turtle programming, but the response for VG was statistically significant. Therefore, VG was more interested in manipulating turtles using programming than TG.

VG and TG both indicated that turtle programming was easier after the lecture, but the results were statistically insignificant. However, from the result of A1, the value of evaluation changed considerably for TG. Therefore, TG tends to feel programming is easier than VG.

VG indicated that the usefulness of turtle programming increased, while TG felt it decreased. The different was significant for VG (A2). The visual expression affects the evaluation as VG intuitively understood the turtle instructions from the illustration.

VG showed an improved willingness to use turtle programming, while TG did not. However, the difference was not significant between the two groups.

Based on these results, VG shows the most improvement, and the results are often more significant than TG. These observations confirm hypothesis H1, which speculates that visual-based programming is adequate for introductory program learning of first learners. In addition, some learners in both groups show reduced values, but the difference is not significant. In particular, many learners in VG commented that the programming difficulty increased after the lecture, whereas many learners in TG indicated a decrease in usefulness and willingness after the lecture. Since TG requires more input, it is possible that the learners had to take their time to program. Furthermore, the degree of difficulty for programming is more likely to change to a positive value for TG.

### *UNDERSTANDING PROGRAMMING*

### Problem results and analyses

We used tests and questionnaires to confirm the comprehension level of first learners. There were six questions (Table 2) and one free problem. Each learner self-declared when a problem was complete and then took a screenshot to confirm the solution. In addition, we acquired the source code as the answer information. Figure 5 shows the response rate. The low response rate was a problem. There was not any difference in P1 by the group. For P2, the percentage of correct answers was higher for VG than TG. This difference is attributed to the amount of input required to program. TG provided a higher percentage of correct answers than VG for P3, which was about loop sentences, indicating that the operation amount (input amount) of VG increased. Consequently, the correct answer rate decreased for VG. The result of P4 was the same as P3. On the other hand, P5, which was about conditional statements, showed the opposite result; that is, VG had a higher percentage of correct responses than TG. Complicated condition expressions had to be inputted for TG. Hence, to obtain a correct response from TG is more difficult than VG. The result of P6 is the same as P5.

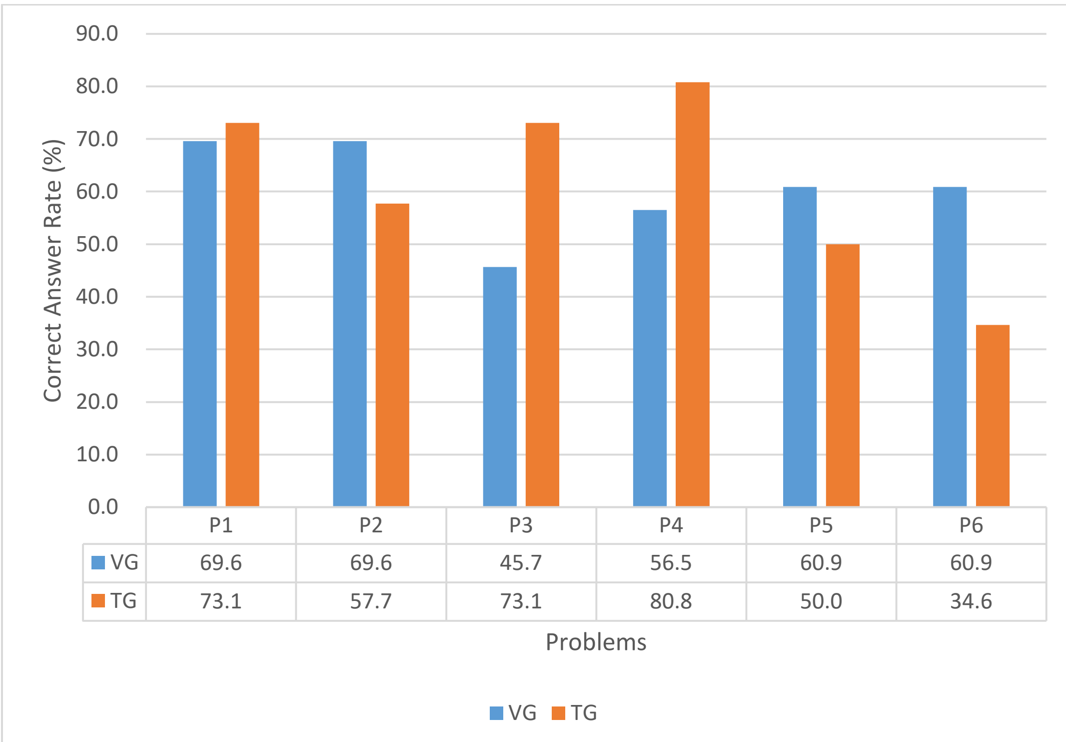



**Figure 5. Correct answer rate**

In addition, the free problem was to create a single letter of the alphabet. Figure 6 shows the answer to the free problem. Both groups utilized many iterations, indicating that a conditional branch is a difficult concept to understand. The results between the groups were statistically insignificant, confirming that the abstraction level of the visual language is similar to that of the text language. However, some learners in both groups were unable to tackle the free problem.

## Description formula questionnaire result and analysis

Q11 and Q12 used the description formula questionnaire (Table 3). Table 7 shows the answer to a questionnaire. The answers were grouped into four categories: "Explain in relation to game events (CTG1)", "Explain the action by words (CTG2)", "Associate with a programming language (CTG3)", "Unanswered · Unknown · Other (CTG4)". "Explain in relation to game events" indicates that an answer was created in association with Minecraft. An example is "Avoid certain blocks the using turtle". Many responses for Q11 and Q12 by the VG group fit into this category. Interestingly, this response was rare in TG. It is possible that VG applied this category more often because the expression of the programming language used for visual input is easy to imagine the event of the game. "Explain the action by words" denotes that the answer was explained using words without relating to game events. In VG, many learners' responses fit into this group for Q11 and Q12. Even in TG, a few learners fit into this group. "Associated with a programming language" represents that the answer was derived from the programming language. An example is "for x = 1, ~ do ~ end". Responses for both groups fit this category, but more responses were from TG. It is possible that TG grasped the meaning of the question as a programming language. "Unanswered · Unknown · others" represents the group that did not respond or indicated that they were unsure. Impressions include, "I do not know" and "It is difficult". This category applied to learners in both groups, but more were from TG. TG may have had more difficulty verbalizing concepts or understanding programming.

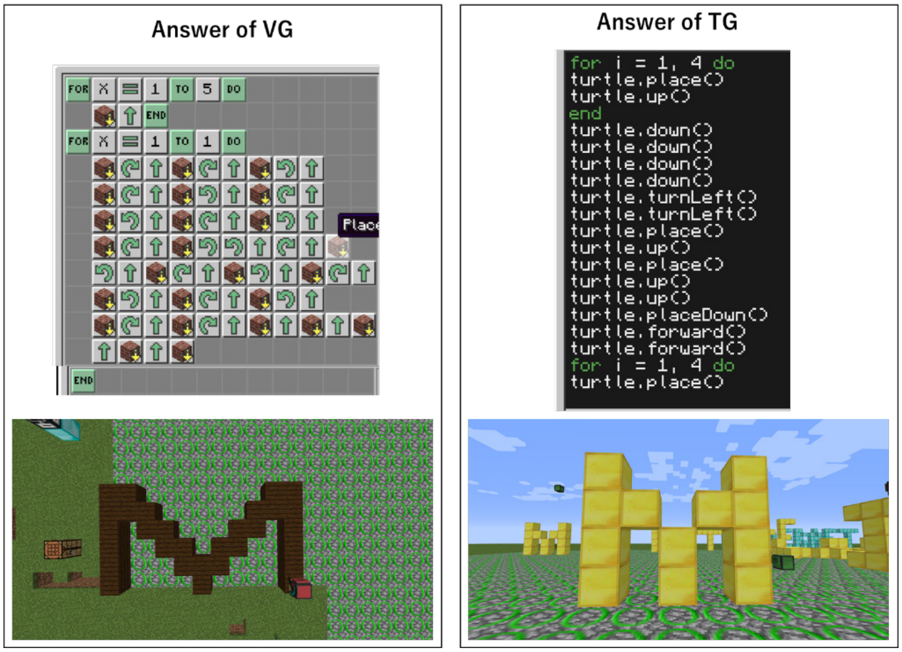



**Figure 6. Answer for the free problem by group**

**Table 7. Applicable rate by category for Q11 and Q12**

| | What is a conditional? | | | What is a loop? | | |
|---|---|---|---|---|---|---|
| | **VG (%)** | **TG (%)** | **Answer Example** | **VG (%)** | **TG (%)** | **Answer Example** |
| **CTG1** | 19.57 | 3.85 | Avoid certain blocks the using turtle | 17.39 | 0 | Process to stack many blocks |
| **CTG2** | 36.96 | 26.92 | If there is ~, run the program. | 34.78 | 15.38 | Repeat as many times as it was said |
| **CTG3** | 13.04 | 23.08 | if ~ then<br>~<br>end | 10.87 | 30.77 | for x = 0, ~ do<br>~<br>end |
| **CTG4** | 30.43 | 46.15 | I am difficult | 36.96 | 53.85 | I do not know |

## Summary of results

The results do not confirm H2. The TG results for the loop problem (Q3 and Q4) were better than those for VG. On the other hand, the results for the conditional problem showed the opposite trend

(Q5 and Q6). In addition, the manipulated variables and input quantities in each input method may influence the correct answer rate.

Q 11 and Q 12 reveal a difference in the type of response for programming concepts according to the programming input method. It is possible that the expression method of the programming language has a significant influence. Since both groups responded similarly to question about the description formula, we believe there is not a substantial difference in the degree of understanding of programming due to the input method.

# DISCUSSION

## *RESULT OF RQ1*

- **RQ1**: Does the visual-based input method induce a different attitude toward programming than the text-based input method?
  - **H1**: The visual input programming lecture induces a larger change in attitude toward programming.

In RQ1, there is a difference between VG and TG for visual expression. The results also differ from previous research. Zorn et al. (2013) used the Mod of CodeBlock for a student lecture course in 2013. Their research, which compared the learning effect of block programming to text programming, found that block programming increases student interest. In our research, VG shows statistically significant differences in "Interest" and "Usefulness" of turtle programming. VG also shows statistically significant differences in "Interest" in programming. These results indicate that visual inputs are likely to increase the interest in programming. Our results provide additional findings. VG also increases "Usefulness" and "Willingness". VG is more intuitive than TG. Because a keyboard was not used in VG, less time is necessary to see results.

TG has no statistically significant difference in some of the analysis. However, the arithmetic mean shows the difficulty programming in TG has a larger improvement than VG. A previous comparison study investigated programming difficulty (Price & Barnes, 2015). According to their research, a novice cannot distinguish the cause of programming difficulties because they do not recognize challenges due to interface differences. However, our study reveals a difference in attitude. This difference may be because the text input is a more realistic programming method than the visual input. Learners may have a prejudice that text is more representative of programming and it is perceived as more difficult. However, the lecture interposed games, creating the possibility that learners felt programming is easy. Hence, programming difficulty TG shows a larger improvement.

The visual input method improves attitudes towards programming more than the text input method. Although the results are not statistically significant, we can confirm H1. Regarding programming attitude for first learners, the visual input method is more suitable. However, the text input method should reduce the difficulty level more than the visual input method. Consequently, the text input method can be adapted to first learners. Accumulating more data in the future should further distinguish between the two input methods.

## *RESULT OF RQ2*

- **RQ2:** Does the understanding of programming between visual-based and text-based input methods differ?
  - **H2:** Programming is easier to understand using a visual input method. (RQ2)

In RQ2, the low response rate is an issue for both groups. VG has a high percentage of correct answers regarding the conditional problem. This is attributed to the fact that the visual method requires less input to create the conditional program. Furthermore, the visual input allows the conditional to be viewed with images instead of text. The score in VG shows a larger improvement than TG. In addition, some learners in VG could not solve the free problem. More answers used loops than conditional branching, suggesting that loops are conceptually simpler than conditional branching. TG has a high percentage of correct answers regarding the loop problem because less input is required to create loops with text inputs. Hence, the score in TG shows a larger improvement than that for VG.

Many responses in VG used the same loop for the free problems. From these findings, it is assumed that both groups are influenced by the operations and input quantity in the environment. In addition, the results also support the notion that a loop is a simpler concept than conditional branching. This result suggests that the expression of programming language influences learners' understanding level if DCT is considered (Clark, & Paivio, 1991; Eitel, & Scheiter, 2015).

Furthermore, the correct response rate for the problem regarding programming indicates that both methods are useful. Similar to above, the results indicate that loops are a simpler concept than conditional. Thus, both methods can be applied to first learners. The programming input method and input quantity may influence the correct answer rate for the problem about the understanding of programming (Table 2). In the questionnaire about programming concepts, VG shows a larger improvement than TG. Consistent with previous research (ChanLin, 2001; Weintrop, & Wilensky, 2015a), this result suggests that is beneficial for first learners to use visual inputs.

The answer to RQ2, H2 cannot be confirmed using the results of this study. The two groups show a clear difference in the understanding of programming. The correct response rate for the problem regarding programming indicates that both methods are useful. Thus, both methods can be applied to first learners.

## CONCLUSION AND LIMITATIONS

We examined whether text input method or visual input method is better for first learners. In the field of DCT and multimedia, it has been reported that visual expressions, as well as applying text and images in a balanced manner, are effective for first learners. Hence, it may be beneficial to teach programming visually. However, programming involves behavioral aspects, such as entering and executing programs. Because information is acquired by more than just site, programming has different aspects from multimedia learning. Furthermore, some studies have applied and compared programming learning methods for first learners, but it is unclear whether a visual input or a text input is more suitable for first learners. Therefore, we investigated the difference in the learning effect of two input methods using ComputerCraftEdu in Minecraft for programming involving first learners. The visual input method results in a larger change in attitude. Significant differences are noted, especially in the interest in programming (including turtle programming). Although the text input seems to make programming less difficult, the difference is not significant.

In the correct answer rate of the problem assessing the understanding of programming (Table 2), there is a difference between conditional branching and loops. The correct answer rate of the conditional problem is higher in the visual input, while the rate of correct answers in the loop problem is higher in the text input. It speculated that the differences are influenced by the operations and input quantity. However, additional studies are necessary to investigate the cause. Differences are found in the questionnaire results of the programming concept. VG tried to explain the concept by applying it to a specific action, while many learners in TG tried to explain the concept in relation to programming. The expression method of programming language may influence the perceptions of concepts.

The overall results indicate that a visual input method is better suited for an introduction to programming. The results coincide with the DCT, implying that it is easier to use a visual input method. However, the comparison results suggest that actions change the learning effect. Hence, the text input method can be used for programming learning of first learners from the viewpoints of the operation amount and input amount in the programming environment. In the future, we plan to investigate the learning effect from the viewpoint of behavior recognition. Furthermore, we plan to collect and analyze additional data as well as determine the correlation between attitudes and understanding of programming.

### *LIMITATIONS*

This research has some limitations. Future research is also mentioned to address these issues. We noted five limitations:

1) The submission rates to the problems confirming the degree of understanding (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 other options.
2) There is a difference in the number of participants in the two groups because participants were recruited via the Internet. Participants selected the group (visual or text) when volunteering for our study. This difference is likely due to the perception that the text method is more difficult at the time of recruitment. Each group should have roughly the same number of participants to address this imbalance in the future.
3) Learners were able to select the programming method because the participants were recruited via the Internet. They could register for either the visual or text lecture. Therefore, participants should be randomly assigned to each method in the future.
4) First learners were recruited online. However, some participants may have had some previous exposure to programming, which may affect the results, especially the understanding of programming concepts. In the future, the filtering and other adjustments will be conducted to reduce the exceptions of participants.
5) The small population size may have affected our results. In the future, more experimental data should be accumulated.

## REFERENCES

Bau, D., Bau, D. A., Dawson, M., & Pickens, C. (2015). Pencil code: Block code for a text world. *Proceedings of the 14th International Conference on Interaction Design and Children* (pp. 445-448). ACM.

Bayliss, J. D. (2012). Teaching game AI through Minecraft mods. *2012 IEEE International Games Innovation Conference* (pp. 1-4). IEEE.

ChanLin, L. (2001). Formats and prior knowledge on learning in a computer-based lesson. *Journal of Computer Assisted Learning*, *17*(4), 409-419.

Chiu, C. F. (2015, April). Introducing Scratch as the fundamental to study app inventor programming. *2015 International Conference on Learning and Teaching in Computing and Engineering* (pp. 219-220). IEEE.

Clark, J. M., & Paivio, A. (1991). Dual coding theory and education. *Educational Psychology Review*, *3*(3), 149-210.

Dann, W. P., Cooper, S., & Pausch, R. (2011). *Learning to program with Alice* (w/CD ROM). Prentice Hall Press.

Du, J., Wimmer, H., & Rada, R. (2016). "Hour of code": Can it change students' attitudes toward programming?. *Journal of Information Technology Education: Innovations in Practice*, *15*, 53-73. Retrieved from https://www.informingscience.org/Publications/3421

Eitel, A., & Scheiter, K. (2015). Picture or text first? Explaining sequence effects when learning with pictures and text. *Educational Psychology Review*, *27*(1), 153-180.

Fraser, N. (2013). *Blockly: A visual programming editor.* Google, Place. Retrieved from https://developers.google.com/blockly/

Gallagher, C. (2014). *An educator's guide to using Minecraft in the classroom: Ideas, inspiration, and student projects for teachers.* Peachpit Press.

Gross, P., & Powers, K. (2005). Evaluating assessments of novice programming environments. *Proceedings of the First International Workshop on Computing Education Research* (pp. 99-110). ACM.

Jiau, H. C., Chen, J. C., & Ssu, K. F. (2009). Enhancing self-motivation in learning programming using game-based simulation and metrics. *IEEE Transactions on Education, 52*(4), 555-562.

Kelleher, C., & Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys (CSUR)*, *37*(2), 83-137.

Kölling, M., Brown, N. C., & Altadmri, A. (2015). Frame-based editing: Easing the transition from blocks to text-based programming. *Proceedings of the Workshop in Primary and Secondary Computing Education* (pp. 29-38). ACM.

Lahtinen, E., Ala-Mutka, K., & Järvinen, H. M. (2005). A study of the difficulties of novice programmers. *ACM SIGCSE Bulletin, 37*(3), 14-18. ACM.

Long, J. (2007). Just for fun: Using programming games in software programming training and education-A field study of IBM robocode community. *Journal of Information Technology Education*, 6, 279-290. Retrieved from https://www.informingscience.org/Publications/216

Mayer, R. E. (2003). The promise of multimedia learning: using the same instructional design methods across different media. *Learning and Instruction*, *13*(2), 125-139.

Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The scratch programming language and environment. *ACM Transactions on Computing Education (TOCE)*, *10*(4), 16.

O'Kelly, J., & Gibson, J. P. (2006). RoboCode & problem-based learning: A non-prescriptive approach to teaching programming. *ACM SIGCSE Bulletin. 38*(3), 217-221. ACM.

Paivio, A. (2013). *Imagery and verbal processes*. Psychology Press.

Price, T. W., & Barnes, T. (2015). Comparing textual and block interfaces in a novice programming environment. *Proceedings of the Eleventh Annual International Conference on International Computing Education Research* (pp. 91-99). ACM.

Saito, D., & Yamaura, T. (2013). A new approach to programming language education for beginners with top-down learning. *International Journal of Engineering Pedagogy (iJEP), 3*(S4), 16-21.

Sáez-López, J. M., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools. *Computers & Education*, *97*, 129-141.

Vasilateanu, A., Wyrazic, S., & Pavaloiu, B. (2016). A science fiction serious game for learning programming languages. *The International Scientific Conference eLearning and Software for Education* (Vol. 1, p. 561). "Carol I" National Defence University.

Weintrop, D., & Holbert, N. (2017). From blocks to text and back: Programming patterns in a dual-modality environment. *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (pp. 633-638). ACM.

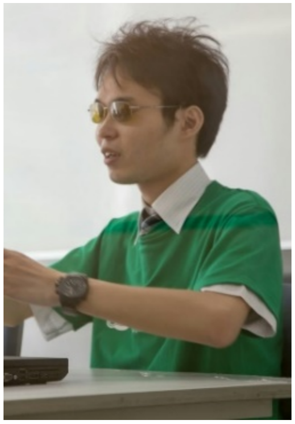
Weintrop, D., & Wilensky, U. (2015a). To block or not to block, that is the question: Students' perceptions of blocks-based programming. *Proceedings of the 14th International Conference on Interaction Design and Children* (pp. 199-208). ACM.

Weintrop, D., & Wilensky, U. (2015b). Using commutative assessments to compare conceptual understanding in blocks-based and text-based programs. *Proceedings of the Eleventh Annual International Conference on International Computing Education Research* (pp. 101-110). ACM.

Wilkinson, B., Williams, N., & Armstrong, P. (2013). Improving student understanding, application and synthesis of computer programming concepts with Minecraft. *The European Conference on Technology in the Classroom*. Retrieved from http://iafor.info/archives/offprints/ectc2013-offprints/ECTC2013_0477.pdf

Zorn, C., Wingrave, C. A., Charbonneau, E., & LaViola Jr, J. J. (2013). Exploring Minecraft as a conduit for increasing interest in programming. T*he 8th International Conference on the Foundations of Digital Games* (pp. 352-359).

## BIOGRAPHY



**Daisuke Saito** is a doctoral course student of the Graduate School of Fundamental Science and Engineering, Waseda University in Japan. He also works as a research associate at the School of Education. He acquired a Master of Information and Telecommunication degree from Tokai University in Japan. His research interests include programming education and digital game-based learning.

**Hironori Washizaki** is head and professor at the Global Software Engineering Laboratory, Waseda University, Japan. He also works at the National Institute of Informatics as visiting professor, and at SYSTEM INFORMATION CO., LTD. as an outside director. He is the Editor-in-Chief of Int. J. of Agile and Extreme Software Development (IJAESD). Moreover, he has served as the editor of various journals including Int. J. Soft. Eng. Know. Eng., IEICE Trans. Info. Sys., Computer Software, and Heliyon. He has contributed to various societies and organizations such as IEEE Computer Society Membership at Large for the Professional and Educational Activities Board, IEEE Computer Society Japan Chapter Chair, SEMAT Japan Chapter Chair, IPSJ SamurAI Coding Director, ISO/IEC/JTC1 SC7/WG20 Convenor, ICST 2017 PC Co-Chair, CSEE&T 2017 PC Co-Chair and APSEC 2018 PC Co-Chair.

**Yoshiaki Fukazawa** received the B.E., M.E. and D.E. degrees in Electrical Engineering from Waseda University, Tokyo, Japan in 1976, 1978 and 1986, respectively. He is now a professor of the Department of Information and Computer Science, Waseda University and the Director of Institute of Open Source Software, Waseda University. His research interests include software engineering, especially reuse of object oriented software and agent-based software.