Picthon

A Learning Environment of Python through Pictogram Content Creation

Kazunari Ito
School of Social Informatics
Aoyama Gakuin University
Kanagawa, Japan
kaz@si.aoyama.ac.jp

***Abstract*—We have been developing an environment for pictogram content creation called "Pictogramming." It also acts as learning the basic concepts of programming languages. This application includes "Pictogram Animation Command" that makes various postures of human-shaped pictograms and "Pictogram Graphics Command" that can draw figures using three different methods. The combination of these two commands will enable a user to create works based on design guidelines in a short period. Pictogramming adopts its own command notation. Hence, we constructed a Python library that can provide the Pictogramming source code as an output and a web-based application that can create pictograms using Python language. We term this learning environment "Picthon." It inherits the all characteristics of Pictogramming. We evaluated the feasibility of class lessons using this application on approximately 130 junior high school students. The questionaire SD (Semantic Differential) scale method is used to analize. We found the result of the lesson which learned by Picthon was well as Pictogramming.**

Keywords—Computer Science Education, Python, Pictogram, Programming

I. Introduction

We developed an environment for content creation, using a human pictogram, called "Pictogramming." Pictogramming is coined from two words, "pictogram" and "programming."

Pictograms are designed and widely used to provide information on the actions or status of a human being. ISO Appendix 3864 includes guidelines on depicting a human-shaped pictogram (e.g. a "human pictogram"). It has primarily been researched in the fields of design and communication areas [1, 2, 3, 4].

Okamoto states that regarding the conceptual understanding of programming, learning effects cannot be expected unless not only the behavior of the output appears as a physical phenomenon but also the learner can recognize it. And stated the importance of a visual manifestation regarding the operation of the output to enhance the conceptual understanding of programming [5], where the evaluation efficiency is based on four statuses: visibility, epicritic, predictability, and independence, which she proposed as guidelines for creating programming teaching materials.

A pictogram is a graphic symbol with high visibility and high epicritic sensation. An animated human pictogram (i.e., a human's motion) can used to represent the output of a program where it is strongly related to an operator's motion(s), and thus the output has a high level of predictability. Moreover, high independence is achieved if the language specification satisfies the fine-grain connections between a human pictogram's motions and the statements within a program.

Using this characteristic of pictogram, Pictogramming also focuses on a learning tool for beginner programmers. In a previous paper, we evaluated the effectiveness of this prototype version of application with approximately 100 junior high school students. The results were generally favorable but it is necessary to improve the content and the method for learning some advanced programming concepts [6]. We also designed lessons for the purpose that integrated design and art education using "Pictogramming", not teaching the programming concepts explicitly [7]. And we analyzed the questionnaires and submitted works. We found many artistic works are strongly linked to one's body movement and self-consciousness. In contrast, design-oriented works are linked to culture, moral, ethics, rule and so on. In other words, a huge difference was seen with free creative work.

Pictogramming adopted unique notation. On the other hand, there is a growing need for education using widely used programming languages. Especially in recent years, Python has been attracting a lot of attention.

In this study, we propose a learning environment called "Picthon," which applies Pictogramming, using the Python programming language.

The rest of the paper is organized as follows: Section 2 explains the background study along with the motivation and related research. Section 3 describes the implementation of Pictogramming and Section 4 proposes the Picthon. Section 5 describes practice and evaluation, and Section 6 summarizes the study.

II. Related Studies

Python is an interpreted language having features such as dynamic typing and multi-paradigms. Libraries are provided in a wide range of fields focusing on science and technology, such as machine learning, data science, and matrix operations.

Many lesson practices using Python have been reported in a wide range of contexts from introduction to programming to the field of artificial intelligence and data science [8, 9, 10]. Dedicated programming tools to learn Python language for beginners have also been developed [11, 12].

In programming education, problems such as lack of interest, difficulty in understanding syntactic rules, and loss of learning motivation due to stumbleness and frustration are considered as problems, and many researches has been conducted to solve these problems [13, 14,15].

Various programming languages have been developed for learning. LOGO is a language that was developed to help children learn various mathematical concepts by operating a turtle robot or turtle character on the screen [16]. Scratch is a

visual programming environment that was developed by the MIT Media Laboratory[17]. Scratch programming is achieved by dropping blocks of code and is suitable for use as an introductory programming course because no specific knowledge of syntax is needed, and no syntax errors can occur. To achieve these advantages, we also released Block Pictogramming[18], block-typed programming version of Pictogramming.

The programming languages such as Scratch mentioned above are focused on user expression and creativity. Under this high degree of freedom, there are many cases for which it is difficult to formulate lessons that set learning goals because they sometimes disrupt learners' creativity.

On the other hand, Pictogramming is designed to use in classrooms, so its command set and functions are limited compared to general programming languages. It also has the advantage that an understanding of the concepts of programming can be achieved through pictogram creation without much learning cost. A learning environment using the Python language can be constructed that can inherit the strengths of Pictogramming. And, a smooth transition from block-typed programming to a general-purpose text programming language is also expected by stepping up using various versions made by common implementation guidelines.

III. Pictogramming

This section explains "Pictogramming."[1]

A. Outline and Web Interface

Pictogramming is implemented as web application with HTML5, CSS, and JavaScript. Figure 1 displays the screenshot. The application comprises three areas: "Human pictogram display area" on the left topmost area, "Program code description area" on the right topmost area, and "Program code assist button area," which is arranged in the form of several buttons to assist code input, at the bottom area. Pictogramming provides following three functions, which are all indispensable for pictogram creation:

1) Posing and animating human shaped pictograms.
2) Drawing objects by line drawing.
3) Displaying safety sign.

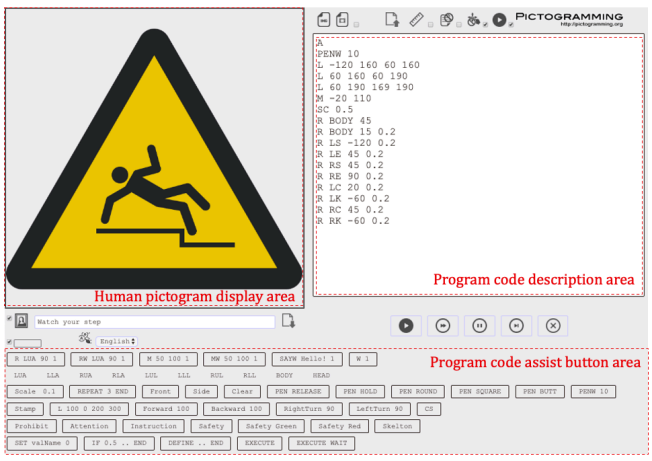

Fig. 1. Screenshot of Pictogramming web interface

B. Coding Specification

Operations on the human pictogram constitute the inputs and are defined in the "Program code description area." Pictogram focuses on pictogram creation and uses a simple and limited instruction set. Figure 2 shows an example of program.

In Figure 2, "`R LUA -120 1`" as illustrated in line 1. "`R`" is an operation to rotate a part of the body specified with the first argument. The human pictogram has a total of nine parts, body and head (considered as a single part), two upper arms, two lower arms, two upper legs, and two lower legs. The body part except for "`BODY`" can be specified by 3 character.

```
R LUA -120 1
RW RUL 10 1
M 100 -20 1.2
IF 0.3
  REPEAT 2
    RW LLA -60 0.3
    RW LLA 60 0.3
  END
END
```

Fig. 2. Sample program

Figure 3 shows the relation between label and a part of body. First character "`L`" or "`R`" indicates left of right side of body. Second "`L`" or "`U`" indicates lower or upper part of an arm or a leg. Third "`A`" or "`L`" indicated an arm or a leg. That is, "`LUA`" means the Left Upper Arm. The second argument gives the degrees counterclockwise of the angle of rotation, and the third argument represents the seconds required to rotate. If the third argument is omitted, then, it would be treated as 0. Hence, line 1 means "Rotate the Left Upper Arm 120° clockwise for 1 second and next command executes simultaneously.

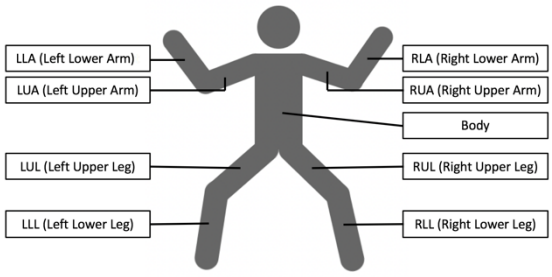

Fig. 3. Shape and name of human pictogram

The character "`W`" in "`RW RUL -10 1`" shown line 2 means that next command executes when this rotation has finished. Line 2 indicates "Rotate the Right Upper Leg 10° counterclockwise for 1 second and the next command is not executed until the movement is complete. "`M`" is an operation to move with parallel displacement. Line 3 indicates "Move 100 rightward and 20 upward for 1.2 seconds." Line 4 "`IF 0.3`" means execute following commands until corresponding "`END`" at a probability of 30%, and Line 5 "`REPEAT 2`" means execute following commands 2 times. Hence, lines 4–9 represent waving the Left Lower Arm (LLA) two times at a probability of 30%.

1 This application can be accessed at https://pictogramming.org/editor/index.html

Drawing figures is essential process of pictogram creation. Pictogramming provides three types of drawing methods [19]. Figure 4 depicts these drawing methods.

In Figure 4 (a), a part of the body is specified and drawn as the movement history of that part. We call this “Human Graphics.” In this example, by drawing a rectangle with the movement of left hand. In this drawing method, commands with time 0 draws a line segment connecting the original position and the position this command establishes. On the other hand, commands with time more than 0 draws the as it is. Various figures can be drawn by combining more than one “`R`”, “`RW`”, “`M`” and “`MW`” commands according to this simple rule.

Figure 4 (b) is a drawing method that shows the movement history of a human-shaped pictogram, which is equivalent to turtle graphics. Hence, we call this “Turtle Graphics.” Many programming language such as LOGO[16], Scratch[17] supports this drawing method. About the transition from graphical language Scratch to text-based language Python by using Turtle graphics is also researched[20]. In Turtle graphics, in the first person viewpoint, a human pictogram is regarded as an agent of the system, in the second viewpoint, the learner identifies itself as a human pictogram, and in the third person viewpoint, the observation and command are assumed to be taken in an interactive manner.

Figure 4 (c) presents the drawing method to draw a line segment in a Cartesian coordinate system. We call this “Coordinate Graphics.” This draws a square with one of the coordinates as (100, 100) and the opposite vertexes as (300, 300). This is a general approach to drawing graphics in programming languages. This is a most major drawing method and is drawn from the viewpoint of the third person.

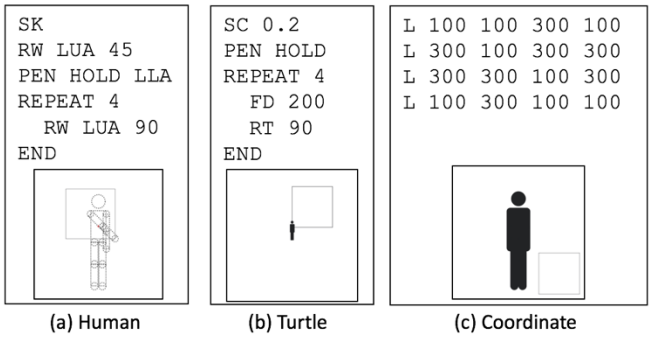

Fig. 4. An example of figure drawing using Pictogramming

Pictogramming also supports to set reference colors, which indicate attention, prohibition, indication, and three kind of safety. Example pictograms adopted reference mark is shown in Fig. 5. All of which could set only single command `P`, `A`, `I`, `S`, `SG`, `SR` respectively.

Fig. 5. Color and shape of safety sign

IV. Picthon

This section explains the programming learning environment "Picthon."[2]

A. Outline and Web Interface

Figure 6 displays the screenshot when accessing Picthon using a PC browser. The application interface is almost same as Pictogramming. The difference point is only Python language is written in “Program code description area” and Python language is inserted by assist buttons in “Program code assist button area”.

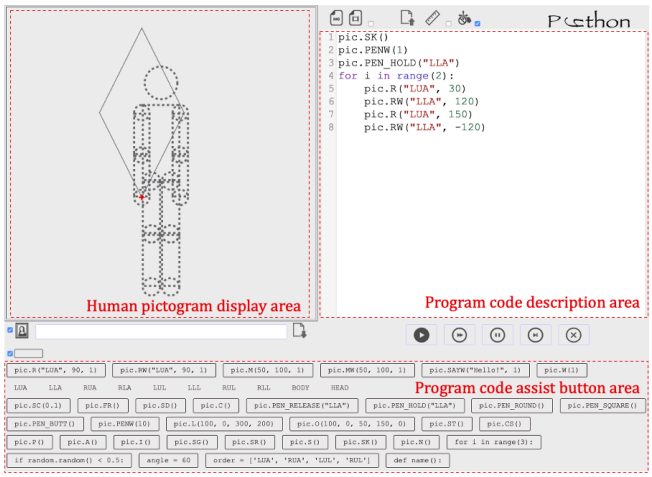

Fig. 6. Screenshot of Picthon web interface

B. Coding Specification and Code Conversion

In this application, the users code using the Python language. Developing a complier to convert between Python and Pictogramming would greatly limit the potential of the Python language. Therefore, we adopted Brython[21], a Python interpreter written in JavaScript, which enable to execute Python program on the browser. Hence, neither a Python interpreter nor any other editors must be installed on the learner’s PC. This can be used in educational institutions where it is difficult to install any native applications to learner’s PC. And we provided a converter Python program, which convert learner’s Python program to output strings that is executable by Pictogramming.

In practice, the converter program does not define the methods that are defined in Pictogramming. However, describing equivalent method missing is feasible in Python. Method missing is a technique that does not generate an error even when the method to be called is left undefined. However, to call a specified method with the method name and arguments, the execution can be performed by generating Pictogramming format commands.

Arguments of Python methods is separated by commas and enclosed by parentheses, but in Pictogramming, as introduced, commands and arguments are separated by spaces.

Figure 7 (a) is a sample Python program on the Picthon application. `pic.r("LUA", -120, 1)` as illustrated in line 1 is equal to `R LUA -120 1` in line 1 of Figure 2.

2 This application can be accessed at https://pictogramming.org/editor/picthon.html

Line 4 is a typical Python '`if`' statement. and line 5 is a loop statement. Hence, lines 4–7 represent waving the Left Lower Arm (`LLA`) two times at a probability of 30%.

For example, the program presented in Figure 7 (a) converts into the program in Figure 7 (b) at a probability of 30% and Figure 7 (c) at a probability of 70%.

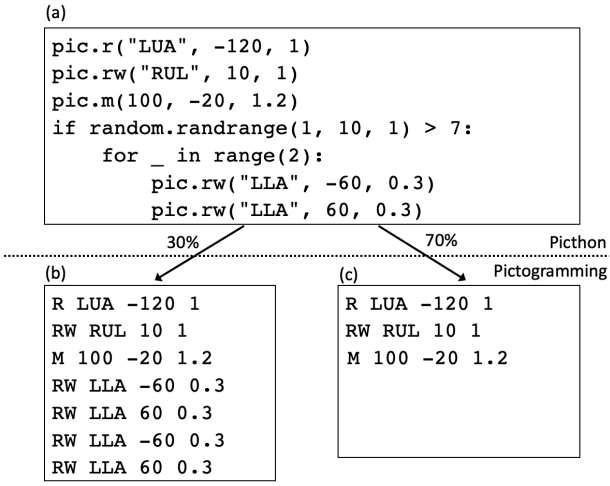

Fig. 7. Example of a Python program and a program that converts into a Pictogramming program

V. Practices and Evaluation

A. Outline

We conducted experiments to evaluate the feasibility of class lessons using Picthon. A total of 131 students were involved, from four classes of 3rd junior high school students. The experiment was conducted in 2019 from June 17, to July 12. Each class lasted for 50 min and was conducted twice a week using standard PCs. Table 1 presents an overview of the experiment. The experiment begins with a 30-minute lecture on the history of pictograms was conducted inspire user interest in pictograms and illustrate how they are used in society. Due to the school events, 2 out of 4 classes were conducted a total of 6 times except the 6th lesson in Table 1, and 2 classes were conducted 7 times as shown in Table 1. And in the second lesson, 2 of the 4 classes were unable to connect to the Internet due to network problems, so the students can not access then Pictogramming program. But they learnt by moving its own body. For this reason, we set up a lesson to review the contents of the 2nd session using a computer at the beginning of the 3rd session.

TABLE I. Overview of the Class Lesson

#	Contents	min
1	1. Short lecture about pictogram and Python 2. How to operate Picthon web application	30 15
2	1. Sequential execution and parallel execution	45
3	1. Sequential execution and parallel execution 2. Looping	10 35
4	1. Variable, list	45
5	1. Human graphics	45
6	1. Turtle graphics, Coordinate graphics	45
7	1. Short lecture about difference between art and design 2. Free Work (from design point of view)	10 35

B. Questionnaire

To evaluate the effectiveness of this method, we provided a questionnaire at the end of each lesson. The questions comprised the eight items mentioned in Table 2. The answer options for Questions 1 to 7 were: "6: strongly agree," "5: agree," "4: somewhat agree," "3: somewhat disagree," "2: disagree," "1:strongly disagree." Q8 was a freeform question. Figure 8 shows the result of the questionnaire for Q1–Q7, and Table 3 depicts the average value for Q1–Q7.

TABLE II. Questionnaire List

Q	Contents
1	You found that the contents of this lesson were fun
2	You can set the theme that you want to make
3	You are aware of the movements of your body during creative process
4	You can make whatever you want to make
5	You are aware of humor during the creative process
6	You are aware of surreal during the creative process
7	You found that the contents of this lesson were difficult
8	Please freely write your comments about this class (free description)

Fig. 8. Questionnaire results

TABLE III. AVERAGE VALUE PER QUESTION

#	Q1	Q2	Q3	Q4	Q5	Q6	Q7
1	5.087	4.500	4.127	4.437	4.833	4.627	3.063
2	5.111	4.825	4.175	4.603	4.746	4.460	3.329
3	5.215	4.752	4.562	4.603	4.851	4.661	3.430
4	4.868	4.388	4.190	4.240	4.512	4.488	4.008
5	5.131	4.777	4.000	4.638	4.938	4.500	3.938
6	4.719	4.328	3.859	4.188	4.438	4.219	4.063
7	5.186	4.853	4.837	4.837	4.853	4.442	3.767

The results for Q1 ("You found that the contents of this lesson were fun") indicate that over 95% of the student responses were rated 4 or above for every lesson except for lesson 4 and 6 and over 90% for lesson 4 and 6. Having fun during practice is a very important element in learning. The data indicates that this condition seems to be satisfied. Since lesson hours are limited, being able to set the theme quickly is an important factor.

The results for Q2 ("You can set the theme that you want to make") indicate a high affinity toward various type of syntonic learning. Although the average values of the 4th and 6th lessons are low, both the lessons include elements such as programming concepts and coordinate systems. This might have been a constraint for the students while having to make free work with these conditions.

The results for Q3 ("You are aware of the movements of your body during the creative process") indicate that the overall average rate for this question is high. However, the values for lesson 5 and 6 are lower than the other lessons. The reason might be the awareness of one's body movements being weakened when the learner focuses on drawing figures with the first person's viewpoint. Further analysis on this is needed in the future.

As for results for Q4 ("You can make whatever you want to make"), being able to make a work in a short time is another important requirement to execute the experiment. The average values of the 4th and 6th lessons are low. It is suggested that the reason for this is same as that of Q2.

The results for Q5 ("You are aware of humor during the creative process.") indicate that over 80% of student responses were rated 4 or above for every lesson. However, the ratio of rate 4 or lower is somehow higher for lesson 4 and 6. There is a precedent study that humor is an element that promotes communication between learners [22].

Surreal is a situation where an expression or idea is unusual or unrealistic. The results for Q6 ("You are aware of surreal during the creative process.") indicate that only the average rate of the 6th lesson is less than the other lessons. In this lesson, students learned Turtle graphics and Coordinate graphics, which made figures without pictograms body movements.

The results for Q7 ("You found that the contents of this lesson were difficult.") indicate that the evaluation is distributed from 1 to 6 for every lesson, and it is considered that all of them have reasonable amount of difficulty.

Table 4 presents the excerpts of the answers to Q8. The following points were identified based on the free commands:

a) According to answer 1 in the first lesson and answer 2 in the third lesson, there were many descriptions of syntonic learning.

b) According to answer 2 in the fifth lesson and answer 3 of the seventh lesson, there were descriptions of social comparisons as one student's work was visible to the other students.

c) According to answer 2 of the fourth lesson, variables and lists were difficult because notations were complicated than previous lessons. Further, some students found turtle graphics was not easy because it used a different command set when compared to the previous lessons.

d) Comments such as "fun," "interesting," "surreal," and "humor" for Q1, Q5, and Q6 were supported. In addition, the comments indicated that many students described the human pictograms as anthropomorphic.

e) The students were conscious of the design-oriented pictograms. They were able to make pictograms based on the design principles of pictograms.

TABLE IV. EXCERPTS OF FREE COMMENTS OF Q8

#	Content
1	Answer 1. It was a lot of fun because I was able to make with my body in mind. Answer 2. I learned that I can make pictograms that I usually look. It was fun.
2	Answer 1. I learned how it works, so it was more interesting than last lesson. Answer 2. I was able to express various movements with characters, and it was interesting to watch the movements that humans cannot animate.
3	Answer 1. Instead of writing many commands to repeat, I felt that it was convenient to use repetition command. Answer 2. I was able to make a running pictogram as if I were a pictogram and was also able to learn while using repetition command.
4	Answer 1. I thought variables would be useful because it enables not to be necessary to rewrite. Answer 2. I understood variables and I was able to use them. But I felt a little more difficult about list because I had to write more instructions.
5	Answer 1. It was fun that I was able to make different diagrams when only angle or time changed. Answer 2. I became interested in how to make them when I saw other's works.
6	Answer 1. It was cute the pictogram became small. I want you to tell me exactly how to make the pictogram. Answer 2. It took time for looking at textbook one by one because there are too much symbols.
7	Answer 1. I saw various pictograms in many places, such as schools, apartments and towns. But I realized that it was necessary to devise something to convey at a glance when I actually make them. Answer 2. I sorrowed to part with the pictogram. Answer 3. I was very impressed by the other's works. I think I learned good things, so I would like to try at home if I have time.

C. *Works Analysis*

In the 7th lesson, following the lecture on different art and design, the students created works on a free theme from a design viewpoint. In this lesson, we discussed that in an artistic works, the creator can freely create and express, and the person viewing can freely interpret it. On the other hand, designated works is something that everyone can accurately interpret.

The analysis was conducted from the perspective of both pictogram animation content and program code. An example of this works is shown in Figure 9.

Fig. 9. Sample of student works

In Figure 9 (1)–(3), the figure is drawn using a pen held by the hand and by moving the arms or body (Human graphics). In Figure 9 (4)–(6), the figure is Coordinate graphics. In Figure 9 (7), the figure drawn is in Turtle graphics. Figure 9 (8) presents the use of all the three drawing methods.

The pictogram is synonymous with "syntonic learning," the importance of which was emphasized by Papert [23]. He covered the concepts of body syntonic (strongly associated with children's senses and knowledge of their bodies), ego syntonic (be consistent with the self-consciousness of children, as a human with intention, purpose, desire, likes, and dislikes), and cultural syntonic (linked to one's activities firmly and positively rooted to one's culture).

Even short practice time within one lesson, various works were created. And, almost all works were related to cultural syntonic. In addition, several of the works in this lesson consisted of the prohibition, caution, instruction, and safety signs defined in ISO 3864 and the theme had the characteristics of body and ego syntonic. Moreover, many students' works were not designed appropriately for everyone to interpret correctly. The consideration of the practice time and definition of "design" have been retained as a future task. From the viewpoint of programming learning, sequential execution, parallel execution, and repetition, "Pictogram graphics commands" are used.

VI. Conclusions

In this study, we constructed a programing learning environment called "Picthon" that expands on "Pictogramming." We illustrated its effectiveness by conducting an experiment on a sample set of junior and high school students and performing an analysis through a questionnaire, user log analysis, and work analysis. We plan to conduct experiments with different environment such as universities. And we will compare with other Python language learning lessons and applications.

References

[1] Y. Mori, T. Takasaki and T. Ishida. Patterns in pictogram communication. In *Proceedings of the 2009 international workshop on Intercultural collaboration* (IWIC '09). ACM, pp. 277-280 (2009)

[2] S. M. Bascón, J. A. Rodríguez, S. L. Arroyo, A. F. Caballero, F. López-Ferreras. An optimization on pictogram identification for the road-sign recognition task using SVMs, *Computer Vision and Image Understanding,* Volume 114, Issue 3, pp. 373-383 (2010)

[3] F. Caffaro and E. Cavallo. Comprehension of safety pictograms affixed to agricultural machinery: A survey of users, *Journal of Safety Research,* Volume 55, pp. 151-158 (2015)

[4] H. Cho, T. Ishida, N. Yamashita, T. Koda, and T. Takasaki. Human detection of cultural differences in pictogram interpretations. In Proceedings of the 2009 international workshop on Intercultural collaboration (IWIC '09). ACM, pp. 165-174 (2009)

[5] M. Okamoto, M. Murakami, N. Yoshikawa, and H. Kita. Development and assessment of learning materials for computer programming focusing on 'visual manifestation', Educational Technology Research, No. 37, pp. 51–60 (2014)

[6] K. Ito. Pictogramming - Programming Learning Environment using Human Pictograms -, IEEE Global Engineering Education Conference (IEEE EDUCON 2018) , pp.134-141 (2018)

[7] K. Ito. A practice and evaluation of "Pictogramming" from the viewpoint of design and art education, IEEE FIE (Frontiers In Education), pp. 1-6 (2019)

[8] Simon, R. Mason, T. Crick, J. H. Davenport, and E. Murphy. Language Choice in Introductory Programming Courses at Australasian and UK Universities. *In Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (SIGCSE '18), pp. 852-857 (2018)

[9] J. Q. Dawson, M. Allen, Alice Campbell, and Anasazi Valair. Designing an Introductory Programming Course to Improve Non-Majors' Experiences. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (SIGCSE '18), 2018, pp. 26-31 (2018)

[10] D. G. Sullivan. A data-centric introduction to computer science for non-majors. In *Proceeding of the 44th ACM technical symposium on Computer science education* (SIGCSE '13). ACM, pp.71-76 (2013)

[11] L. Grandell, M. Peltomäki, R. Back, and T. Salakoski. Why complicate things?: introducing programming in high school using Python. In Proceedings of the 8th Australasian Conference on Computing Education - Volume 52 (ACE '06), Denise Tolhurst and Samuel Mann (Eds.), Vol. 52. Australian Computer Society, Inc., pp. 71-80 (2006)

[12] P. J. Guo. Online python tutor: embeddable web-based program visualization for cs education. In Proceeding of the 44th ACM technical symposium on Computer science education (SIGCSE '13). ACM, pp. 579-584 (2013)

[13] J. Helminen and L. Malmi. Jype - a program visualization and programming exercise tool for Python. In Proceedings of the 5th international symposium on Software visualization (SOFTVIS '10). ACM. pp. 153-162 (2010)

[14] E. Lahtinen, K. Ala-Mutka, and H. Järvinen. A study of the difficulties of novice programmers. In Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education (ITiCSE '05), pp. 14-18 (2005)

[15] C. Kelleher and R. Pausch. Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Comput. Surv.* 37, 2, pp. 83-137 (2005)

[16] P. Seymour. A learning environment for children., Computers and Communication: Implications for Education, New York: Academic Press, pp.271-278 (1977)

[17] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman, and Y. Kafai. Scratch: programming for all. Commun. ACM 52, 11, pp.60-67 (2009)

[18] K. Ito, Block Pictogramming -A Block-based Programming Learning Environment through Pictogram Content Creation-, IEEE Global Engineering Education Conference (IEEE EDUCON 2020) , pp.1669-1673 (2020)

[19] K. Ito. Figure Drawing Method Based on Human Motion using Pictogramming, IEEE TALE (An International Conference on Engineering, Technology and Education) , pp.1-6 (2019)

[20] M. Dorling and D. White. Scratch: A Way to Logo and Python. In Proceedings of the 46th ACM Technical Symposium on Computer Science Education (SIGCSE '15). ACM pp.191-196 (2015)

[21] Brython. http://www.brython.info/

[22] J. W. Neuliep. An examination of the content of high school teachers' humor in the classroom and the development of an inductively derived taxonomy of classroom humor, *Communication Education*, vol. 40, no. 4, pp. 343–355 (1991)

[23] P. Seymour. Mindstorms, Children, computers, and powerful ideas. Basic Books, Inc (1980)