

Programming in Japanese for Literacy Education

**Ken Okada, Manabu Sugiura, Yoshiaki Matsuzawa,
Megumi Araki, and Hajime Ohiwa**

Keio University - Graduate School of Media and Governance
5322 Endo, Fujisawa-shi, Kanagawa-ken, 252-8520 Japan
squeaker@crew.sfc.keio.ac.jp

Abstract: We have developed a programming language Kotodama (means “what you speak becomes reality”) in which a source program can be read as authentic Japanese language and be executed as a programming language at the same time. We put Kotodama into a teaching environment Squeak developed by Alan Kay and have developed several programming courses on it. We have found that explanation of a programming language becomes unnecessary and that we can concentrate on algorithm development by examining a program text that can be read as Japanese. We believe that this programming course can work as natural language writing course as well for precise description.

Keywords: Programming in Japanese, Word order, Postfix notation, Algorithm education

1 Introduction

When personal computers appeared, a programming language BASIC first appeared and users would learn programming by using it. However, various software applications had been commercially available and it soon became unnecessary for users to learn programming.

We believe that leaning how to use application software does not form a literate person for information society. Instead, we believe teaching the essential understanding of the computer by programming is to form literacy for the people in information society.

At first, the work of programming occurred in assembly language. It used mnemonics that corresponded to the machine instructions. Then, high-level

Please use the following format when citing this chapter:

Okada, K., Sugiura, M., Matsuzawa, Y., Araki, M. and Ohiwa, H., 2008, in IFIP International Federation for Information Processing, Volume 269; *History of Computing and Education 3*; John Impagliazzo; (Boston: Springer), pp. 171–176.

programming languages such as FORTRAN appeared. Programming languages after FORTRAN were designed as abbreviated languages of English.

Research on machine translation (MT) has long occurred but few reported its application for programming languages. As is well known, MT is useful when the scope of translation is limited to the narrow area. Programming would be one of the most suitable areas of MT application.

Squeak by Alan Kay is an excellent environment for teaching programming, and people reported many good results for introductory programming education. However, when applied for the Japanese, it does not work as it does for Western people. This is because a program appears in the form of the English language, as is explained in the next section. We extended Squeak eToy so one can write a program in the Japanese language; we named it “Kotodama on Squeak” to solve the problem.

2 Word Order

In Squeak, a program is composed of by arranging scripting tiles so that the composed tile script becomes a program based on English. Figure 1 is a tile script for commanding the star forward by five dots. In this expression, the subject comes first, the verb comes next, and the object comes at the end.

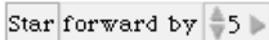


Figure 1 An example of Squeak command in English

In Japanese, the subject comes first, the object comes next, and the verb comes last. The word order “object verb” is same as that of the postfix notation and very effective for stack operation. The word order of “Kotodama on Squeak” is different from English word order. In Japanese, Squeak (Squeak Nihongo) in which an English word is replaced by the corresponding Japanese word, the verb comes next to a subject.

“Kotodama on Squeak” enables one to write a program in Japanese. Figure 2 is a tile in “Kotodama on Squeak”. The verb is located at the end, following Japanese word order.

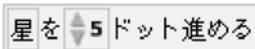


Figure 2 Kotodama on Squeak command in Japanese

It is very difficult to understand a tile script of the replaced words in the Japanese Squeak. To illustrate this, Figure 3 shows the translated English script of the tile script of Figure 2 without changing the word order. (“[p]” shows the

particle). Native English people would feel strange to see the verb that comes at the end. It is what happens to Japanese users while they use the Japanese Squeak.



Figure 3 Translated English command of Figure 2 with the same word order

Let us imagine that we use the tile script shown in Figure 3. An instructor must explain that its meaning is “Star forward by 5”. This is what happens when we use Japanese Squeak.

When we use “Kotodama on Squeak”, the instructor would say that one must just read the tile script and think what is wrong with it, if a learner realizes that something is wrong because his/her program does not work as expected.

3 Tiles Whose Meaning can be Understood by its Reading

In Squeak eToy, tiles exist that are not easy to understand for the novices. The test tile shown in Figure 4 is one of them.



Figure 4 Test tile in Squeak

It is difficult for the beginning programmers to understand the meaning from this. Only those who know the concept of condition branching would understand the meaning of “Test”, “Yes”, and “No”.

In “Kotodama on Squeak”, the corresponding tile script is understood only by reading the tile. Figure 5 shows such a test tile.

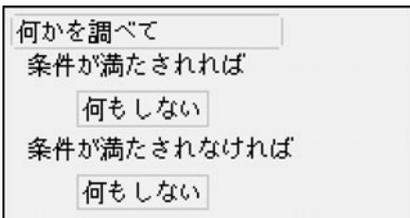


Figure 5 Test command in Kotodama on Squeak

Figure 6 shows the tile in which the Japanese text of Figure 5 translates into English.

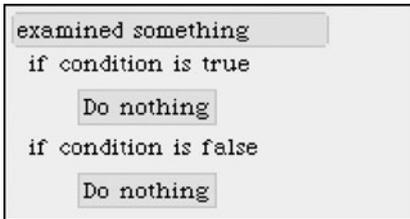


Figure 6 English translation of Figure 5

In “Kotodama on Squeak”, a tile shows such script as the one which even a program beginner understands the meaning by reading. When one reads, “examined something”, they immediately understand its meaning. In addition, when we read the expressions under it, we will also understand their meanings.

Good readability of the tile in “Kotodama on Squeak” occurs when we place some tile on the conditional part of the test tile. Figure 7 is such an example in which the color judgment tile appears on the conditional part. Figure 8 shows its English translation.

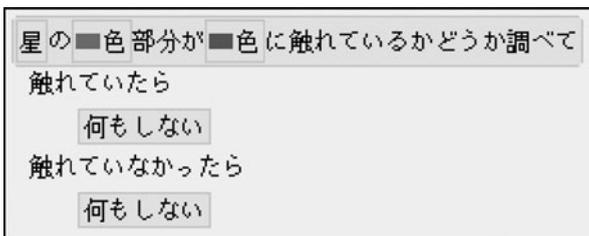


Figure 7 What is to be done is described in Japanese in Kotodama on Squeak

In the test tile of “Kotodama on Squeak”, when we make some condition, the expression of the corresponding second and fourth lines change according to the content of the condition as is shown in Figure 6 and Figure 8. Thus, the instruction of Squeak becomes easy because it uses scripts that we can understand by reading them, as we did in the last section.

Because it is difficult for novices to grasp the meaning in the test tile of Squeak eToy, its instructor has to explain the meaning. If we use “Kotodama on Squeak”, the instructor’s explanation becomes one such as “read the tile as it is and think what it means”. Eventually, learners start to consider the meaning of the tile by reading it for themselves while variously changing it for reaching the desired result. “Kotodama on Squeak” promotes the learners’ self-study attitudes.

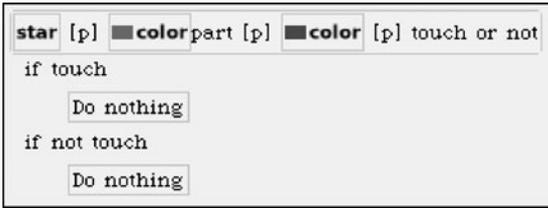


Figure 8 English translation of Figure 7

4 Interface to Understand the Operation Method

In Squeak eToy, an interface requires explanation to understand its effects. Figure 9 shows such an interface when we combine the variable tile and the assignment statement tile.



Figure 9 Prepared tiles for assignment

It is very difficult for a beginner to remove the tile as is expected from the interface. We can take out the assignment statement tile by clicking the arrow image. Without instruction, it is difficult to form this operation. It is necessary to click the arrow part of the assignment statement tile to obtain the increase type tile as shown in Figure 10. This operation is also difficult for novices.

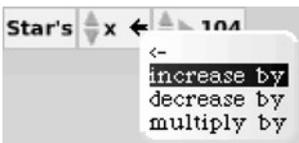


Figure 10 Increment statements for the variable “Star’s” in Squeak

We can modify the interface for manipulating the variable in “Kotodama on Squeak”. We need one such modification because the current value of a variable is always displayed. It is difficult for the programming beginners to understand the concept of the variables. Therefore, by always displaying the current value, we force the learners to be conscious of the variable and its current value.

Another modification is to separate the assignment statement tile and the increase type tile. We do this because the meaning of the object in the sentence is

different in the former and latter statements. For example, when writing “Star’s x set to 5”, the “5” means the value of x to be set after the assignment. When writing “Star’s x increase by 5”, the “5” means the amount of the change of x . We denote notations for setting and arithmetic operations differently in Japanese.

There is a disadvantage that complete Japanese notation needs more screen space, but “Kotodama on Squeak” gives priority to comprehensiveness. Although we use one line of space for showing one variable in Squeak eToy, three lines are necessary in “Kotodama on Squeak”. Therefore, we install the geometry category by using three screens.

The interface design philosophy of “Kotodama on Squeak” is that one can understand the meaning of a program just by reading it. Therefore, together with the features of preceding sections, the learner’s self-study attitude is advanced.

5 Conclusion

“Kotodama on Squeak” is a programming environment intended for introducing programming for Japanese novices. We provide several courses for K-6 to K10 and university students. Learners can understand the effects of a program by just reading the source program as a Japanese text. We have found that education for algorithmic developments can be taught in shorter hours than the one taught by using conventional programming languages, because grammatical explanations are unnecessary. The user-friendly nature of Squeak also contributes to successful educational results. We find that using natural language is very effective for teaching programming. It could be useful for precise description of a natural language as a part of traditional literacy education.