

Computer Programming Education Using the Visual Programming Language Viscuit

Yasunori Harada[†]

Abstract

This article introduces the visual programming language called Viscuit, describes how it is used in practice to promote computer programming education, and explains why programming education is necessary.

1. Introduction

Computers and the Internet have brought big changes to the world, and along the way the technology that has contributed the most to those changes is programming. The changeable nature of programs for servers and clients is the main reason the Internet can provide various information flows and enhance the freedom of communication. Furthermore, the continual advances being made in programs for computer graphics let us watch spectacular movies and special effects. The power of computers arises through programming, and it can be said that the essence of computerization is programming.

It has often been thought that only persons with special skills or knowledge can program. As the application range of computers expands, a division of labor has arisen, where one person gets the idea for an application and another person writes the program for it, with the actual programming relegated to the project's lowest priority. In other words, programming is taking longer and longer in spite of the steady infiltration of computers into our lives.

Programming is important from the standpoint of literacy education. Children intuitively understand dynamics. For instance, intuition about dynamics tells children that if they stack blocks in certain way, the stack will collapse. Rules restricting actions are

not needed when we have intuition about the laws of physics.

On the other hand, when we do not have intuition about the laws of physics, we must follow rules. For instance, consider a storeroom in which loads are stacked. To prevent a collapse, the rule in this case might be that four loads can be stacked. But if we have intuition about the laws of physics, we would never stack loads too high in the first place. That is to say, we would not need stack height rules.

In using computers, what are the consequences of not knowing the fundamental laws of computerization? There are a lot of rules that we must obey to use a computer. Because we lack intuition about the fundamental laws, we must remember a lot of rules. If we forget the rules, a file may vanish, or even worse, considerable inconvenience may result. Moreover, the rules change over time. Considering the ever-changing nature of software, operating systems, and computer hardware, it would be worthless to learn the rules by heart. Education about the fundamental laws of computerization is needed. Knowledge of these laws reduces the number of rules that must be learned to use a computer safely. Because programming is the main characteristic of the computer, programming education will help learners grasp the fundamental laws of computers.

2. Viscuit

Viscuit [1]–[3] is a visual programming language

[†] NTT Communication Science Research Laboratories
Atsugi-shi, 243-0198 Japan

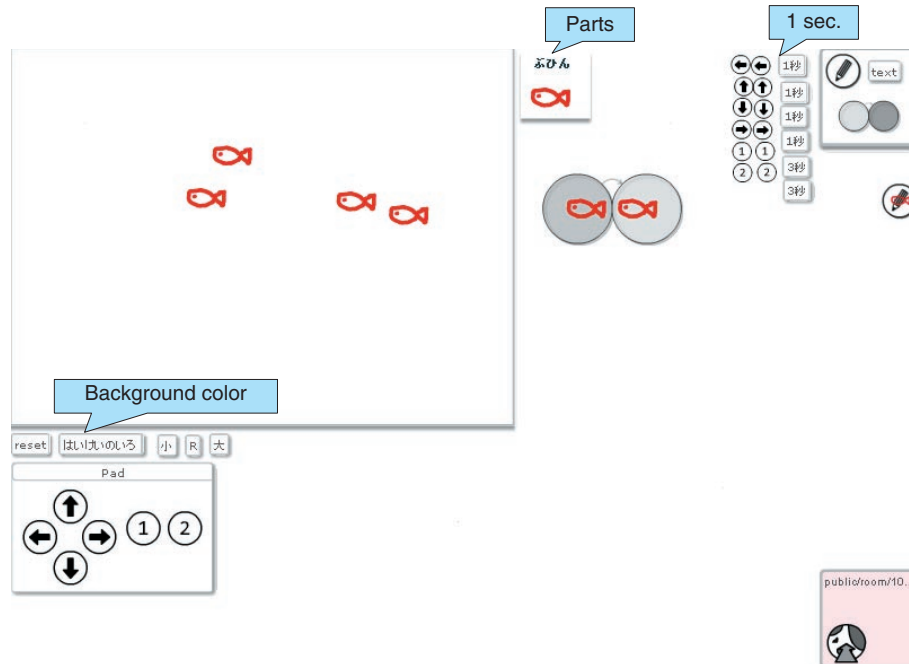


Fig. 1. Basic screen of Viscuit.

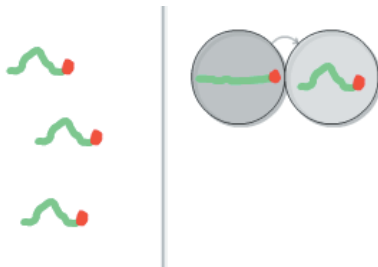


Fig. 2. Picture drawing (1).

for manipulating pictures on the basis of rules. The interface was designed for small children (there are no menus, right-mouse-button operations, or double clicks), but an adult can use it without stress. An example of how Viscuit is used is shown in **Fig. 1**. When we draw a picture, the picture becomes something that can be used by a program. We represent a program by a pair of eyeglasses. First, we drag several parts (e.g., fishes) onto the stage (upper right in Fig. 1). Then, we drag a pair of glasses and drag two parts into its lenses. The change from the left lens to the right lens represents a temporal change for the parts on the stage. The scene in the left lens is replaced

by the scene in the right lens. Since the fishes in the left and right lenses are in different relative locations, this program causes the parts on stage to move. The fish on stage is replaced by one slightly further to the left, so it appears to move ahead from right to left.

In the glasses in **Fig. 2**, the left lens has a straight worm and the right one has a bent worm. This means that a straight worm turns into a bent one. The straight worm on the stage becomes a bent one. What happens when we add other worms onto the stage. When we drag a straight worm onto the stage, it immediately changes into a bent one. But when we drag a bent worm onto the stage, it does not change. So there are many bent worms and no straight ones on the stage.

When I ask children what we can do to straighten the bent worm, many get the idea of exchanging the right and left parts in the glasses. In **Fig. 3**, the bent worm is on the left and the straight worm is on the right. Sure enough, the bent worms immediately become straight. Next, I change the left and right parts of the glasses again and the straight worms become bent. Sometimes, I repeat this. Changing the left and right parts in the glasses causes the worms to change shape: bent or straight.

At this point, I ask what I should do so that the worm stretches out and contracts continuously?

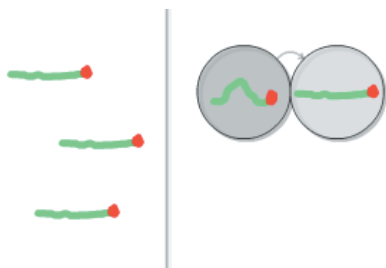


Fig. 3. Picture drawing (2).

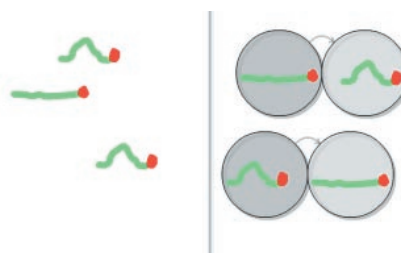


Fig. 4. Picture drawing (3).

Several children can arrive at the correct answer after about five minutes of trial and error. Moreover, with a hint (I show them two pairs of glasses), most children can get it. The correct answer is to arrange two pairs glasses, as shown in **Fig. 4**. The worms are animated by stretching and contracting. When other worms are added, they are also animated. I think that there is a barrier between the case of one pair of glasses and the case of two pairs. The former is just a command for controlling the computer. Children familiar with machines and devices know that pressing a switch will activate it (e.g., a wireless remote controller). When we operate such devices, something moves or changes in response to the operation. In other words, the operator is issuing an order. The latter is a definition of the computer's behavior. When children arrive at the correct answer, the worms begin to animate. This means that they have programmed their computers. The glasses become the description of the movement rather than the simple order: "If a worm is bent, then it straightens; if a worm is straighten, then it bends." And the children know that the program has a meaning: the machine moves automatically by means of a description of the movement rather than working by this human order.

This is the first programming experience for these children. Next, they learn about improving a program. In fact, the worm cannot move forward properly, but merely stretches on the spot. They need to adjust the relative positions in the lenses. They can make a worm to go forwards or backwards by adjusting the relative positions of its pictures.

Some children who understood this concept undertook the challenge of making a three-frame animation. In the three-frame animation with three pairs of glasses, the three parts (e.g., A, B, and C) must be made to change in turn (e.g., A->B, B->C, and C->A). The children guessed the movement that the three frames would produce without learning it and they

succeeded in making an animation. When the computer was regarded logically in this way, the children were able to learn that it was a partner that they could easily work with.

A more advanced example is shown in **Fig. 5**. The upper glasses mean that the fish goes ahead (just as in Fig. 1). The lower glasses (newly added) mean that when a fish hits seaweed, it avoids the seaweed. The fish on the stage swims forward and avoids seaweed. This example shows a complicated programming condition. A function playing a key role in this language is *soft-rewriting* [4]. It matches the placement of the picture flexibly and rewrites it depending on the gap. This function makes Viscuit very soft (flexible) and a highly expressive programming language.

A game that one fourth grader made is shown in **Fig. 6**. Using a game pad and keyboard operation, we can switch the glasses on and off. The goal of the game is to avoid hitting a wall or obstacle. Although the boy who made it had no computer skills when he started, he was able to make this game in only 200 minutes. He got the idea for the game from other children and he gave the game various degrees of difficulty. He learned programming skills in a very short time and was able to express himself.

The games that can be made with Viscuit are limited. Because there are no variables in a Viscuit program, it is very hard to use numerical values, such as keeping score in a game.

3. Hands-on workshops

In 2009, I conducted nearly 50 workshops around Japan at about 30 different locations. I kept the following ideas in mind.

- (1) Organize the workshops so that participants show things to each other.

When children understand information that is useful to themselves, they think that it will also be

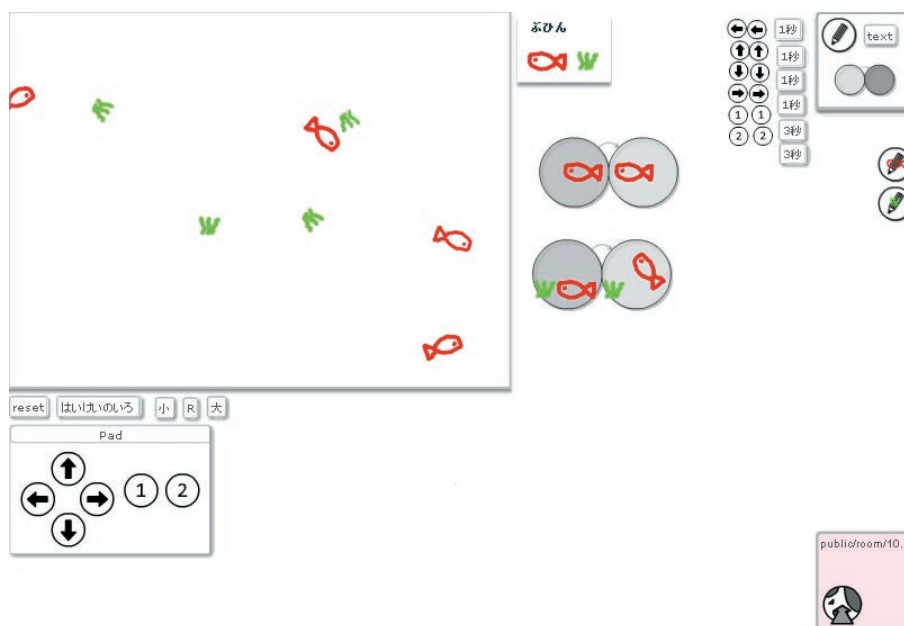


Fig. 5. Condition estimation.

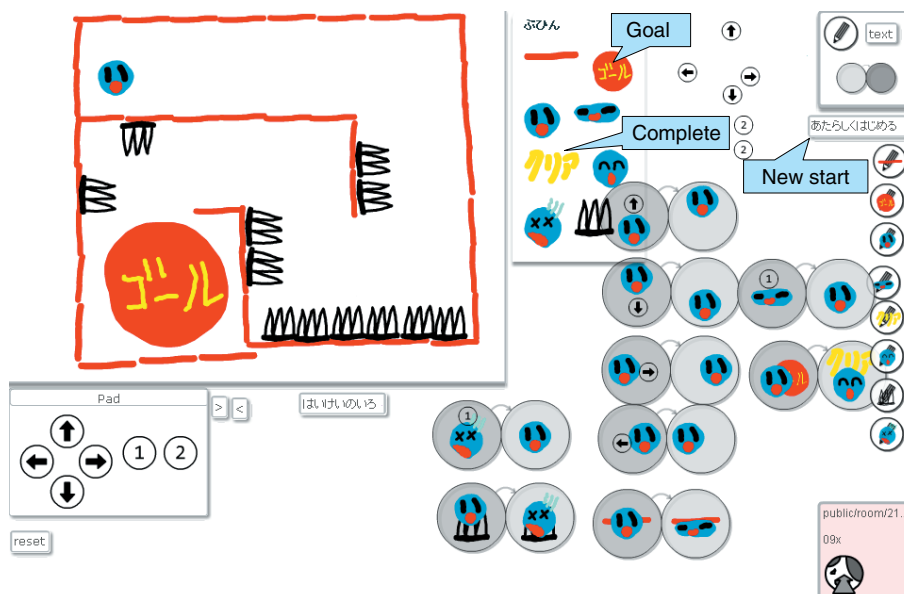


Fig. 6. Game made by a fourth grader.

useful to other people. Thus, we often observe children teaching each other. If the participants have not yet established friendships, we must locate their desks and computers carefully to enable natural conversation.

(2) As much as possible, do not teach.

Because the functions of Viscuit are easy to find, children can discover them by themselves by trial and error. Education based on letting children discover things by trial and error on their own is ideal.

Computers offer many advantages in this type of education because they allow repeated trial and error safely.

(3) Open the door to the joy of mathematics.

Some children want to control the animations precisely. The glasses have a hidden button permitting this, which we show only to children who demand greater control of the movement. When this button is pressed, a part's location is expressed with a numerical value. We give these children the numerical value expression and this increases their interest in mathematics.

(4) Explaining to parents is important.

The importance of programming is often not fully grasped by adults. We therefore explain Viscuit carefully to parents observing the workshop.

4. Need for programming education

The workshop is not only for children; it is also for university students and adults. The reactions of participants reveal that there is a barrier to programming in all generations, which means that they cannot

acquire programming skills naturally through daily living.

In education, the importance of 21st Century Skills [5] (e.g., critical thinking, communication skills, and autonomous learning) is often debated. Such skills are undoubtedly important, but programming is a skill that cannot be acquired naturally. I think that, for future prosperity, everyone should have an intuitive understanding of programming.

References

- [1] Viscuit website (in Japanese).
<http://www.viscuit.com/>
- [2] NTT Group Communication EXPO—R&D Exhibit Report, NTT Technical Review, Vol. 4, No. 5, pp. 14–21, 2006.
<https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr200605014.pdf>
- [3] Y. Harada, "Software Development for Hands-on Workshops," Proc. of the 50th Programming Symposium by the Information Processing Society of Japan, Kanagawa, Japan, 2009 (in Japanese).
- [4] Y. Harada and R. Potter, "Fuzzy Rewriting—Soft Program Semantics for Children," Proc. of the 2003 IEEE Symposium on Human Centric Computing Languages and Environments, Vol. 1, No. 1, pp. 39–46, Auckland, New Zealand, 2003.
- [5] <http://www.p21.org/>



Yasunori Harada

Senior Researcher, Communication Environment Research Group, Media Information Laboratory, NTT Communication Science Laboratories.

He received the M.E. and Ph.D. degrees in information engineering from Hokkaido University in 1989 and 1992, respectively. He joined NTT Basic Research Laboratories in 1992. He was a researcher in the Japan Science and Technology Agency from 1998 to 2001. He was a project manager in the Information Technology Promotion Agency of Japan from 2004 to 2006. He started to develop Viscuit in 2003.