

# The Practice of Teaching Java Language to Visually Impaired Students

Nobuyuki SASAKI

Tsukuba University of Technology

**Abstract:** This paper describes teaching the Java language to visually impaired students. We focus specifically on how to help blind students understand graphic objects and GUI tools and how to help them feel at ease about programming. We applied three main practices. The first was an animation application using Java Applet programming and a tactile graphic device. A very simple robot animation was generated by thread programming. The second was a synthesized speech guide to recognize mouse movements. A jukebox application was used as an example. Finally, the third was a pre-processor system for editing source codes for typical processes to reduce the total statements. In each case, we obtained good results. The blind students understood the output of their program, which had previously been difficult to confirm, and they felt at ease about writing source codes. Eventually, they felt very enthusiastic about taking on complicated programming work through these three practices.

**Key Words:** Java, tactile device, visually impaired students

## 1. Introduction

Java is a very good language for teaching computer programming to visually impaired students for the following general reasons:

- Easy design and maintenance because Java is object-oriented.
- Java is network-oriented and platform-independent.
- Programming is easy because the grammar is based on C language without pointers.
- Multimedia handling is easier using Applet programming compared with other languages.
- Real-time processing is easily implemented because multithreading is available.

The CUI-level human interface for the development environment is very useful for visually impaired students who cannot take advantage of visual development tools. However, there were several problems in teaching the Java language:

- Confirming the results of computer programming in the graphical application of Applet programming is difficult, so the target has been text-based programming thus far.
- Visually impaired students cannot handle low-level interfaces such as mouse devices.
- Some coding tasks become more complicated than with C language, such as handling keyboard input or file I/O. Further, the fact that many more statements are needed makes it very hard for visually impaired students who can recognize the source code only by synthesized speech.

The author began teaching the Java language to visually impaired students three years ago and has been using several techniques to resolve the abovementioned problems. In this paper, the following techniques and results of practice are reported.

- Use of a tactile graphic display to test the graphical application.

- Use of a synthesized speech guide to recognize the mouse movement.
- Development of a pre-processor for editing source code for a typical process in order to reduce statements.

There has been some related research [1][2], but very basic and concrete practices in real lessons—recognizing animation, experiencing the low-level interface of mouse operation, and developing the pre-processor to reduce the total source code—have not been previously reported.

## 2. Use of a tactile graphic display

Three years ago, a tactile graphic display (DV-2) was introduced at my university. This pin display's resolution is 32\*48, all or part of the active window can be used. The minimum refresh interval is 0.05 second, so real-time displays can be achieved. Fig. 1 shows the exterior of the DV-2.



Fig. 1. Exterior of DV-2

Using this device, four practice methods were executed. These are described in the following sections, along with the results.

- Graphic displays of plural figures, such as lines, rectangles, and circles (some of them were filled): The students made the program and recognized easy combinations of two or three figures, but could not recognize complicated combinations of more than four figures. Fig. 2 shows the display view of the Applet of plural figures that were used for recognition.
- Animation of binary counter: A two-bit binary counter source code was shown. The students could understand the motion of the binary counter, just like sighted students do with LEDs, and they could modify the program to make a three-bit binary counter. Fig. 3 shows the display view of the Applet of two-bit binary counter.
- Gif animation: The students tried to recognize a gif animation from a commercial greeting card website, but they had difficulty recognizing the contents. Fig. 4 shows the Gif animation that was used.

Animation of a robot's face and body: These features were drawn by combining lines, rectangles, and circles. Thread programming was used for changing the face and body at certain intervals (e.g., 500ms). Every student made a program and evaluated both their own program as well as the other students' programs. Fig. 5 shows the display view of the Applet used for the robot face animation.

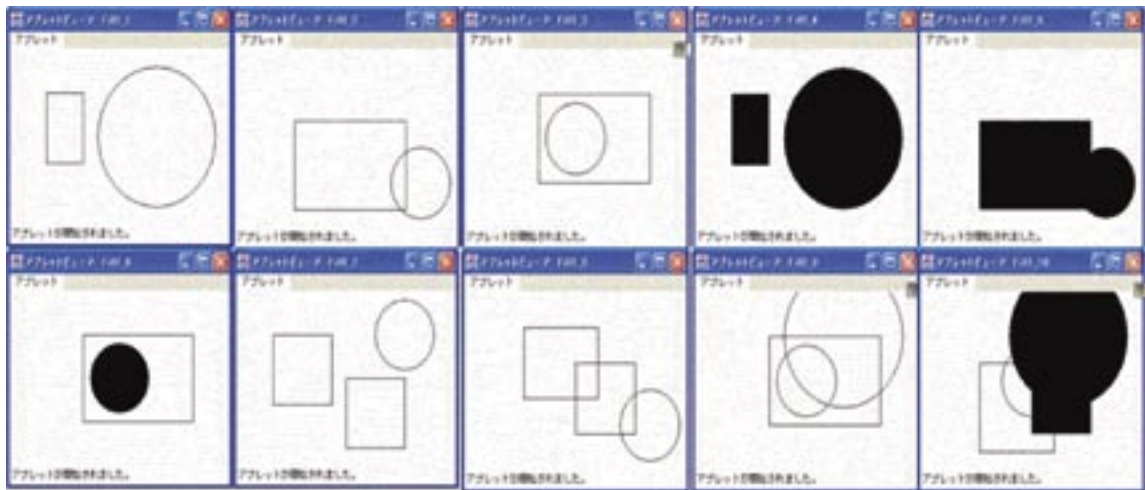


Fig. 2. Multiple figures for recognition

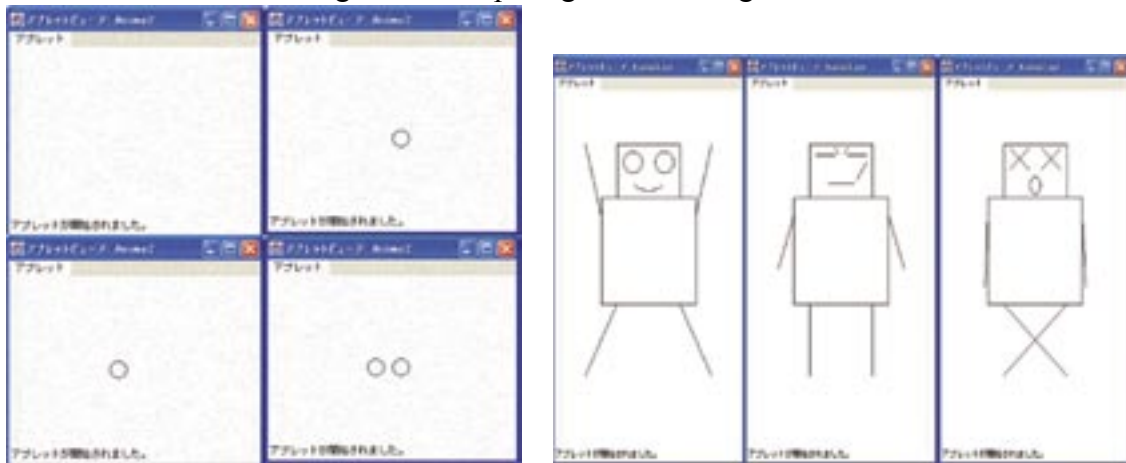


Fig. 3. Two-bit Binary Counter Animation

Fig. 5. Animated Robot Face and Body



Fig. 4. Example of Gif animation

During these exercises, a tactile display proved to be a strong tool for helping visually impaired students recognize graphic applications, much like a LCD display is used by sighted students. A tactile display is limited in the graphic information that it can show because the resolution is much less than that of an LCD display, but when the students used it skillfully (ex., for hierarchical use), it worked well.

### 3. Use of synthesized speech guide to recognize mouse movement

Visually impaired students normally use short-cut keys instead of a mouse. However, it is important for them to experience using a mouse; therefore, the exercises for the “Jukebox application” were adopted. In this application, the window is divided into  $n*m$  blocks. In each block, when a right mouse button is clicked, the music title corresponding to the position of the pointer is announced in a synthesized speech message. When a left mouse button is

clicked, the song corresponding to the position of the pointer is played. When the pointer exits or enters the window, a corresponding synthesized speech message (“exited” or “entered”) is played. The students studied the source code of the “Jukebox application” and modified it by adding several functions, thereby extending the “Jukebox application” class. All of the visually impaired students who did not have experience operating a mouse learned how the mouse works.

#### **4. Development of a pre-processor for editing source code to reduce statements**

Although Java has a lot of merits, it is object-oriented and needs much more source code than C language for a typical process such as inputs or outputs. Usually, visually impaired students recognize the source code through synthesized speech, so it is very hard for them to recognize the total program as the volume of source code increases. Therefore, the macro instructions for a typical process such as file or device inputs and outputs were introduced, and the pre-processor that converts macro instructions into detailed source code was developed. Using this pre-processor, if students use macro instructions for file or device input or output processes, they can reduce the source code, which helps them to gain a clearer view of the whole program they wrote. This practice is being tried now and many visually impaired students feel it helps them with the programming.

#### **5. Conclusion**

Java is an appropriate programming language for visually impaired students, but there are several problems. These are a graphical application, using low-level interface and input and output processes. The author tried several techniques to resolve these problems, and for all of the items the visually impaired students did learn Java successfully. In the future, a total development environment of Java language for visually impaired students will be developed combining graphical applications, low-level interfaces and input and output processes.

#### **References**

- [1] Suguru Yamamoto, Hideji Nagaoka and Yutaka Shimizu: Development of a system to assist Java programming for severely visually impaired people, Human Interface 2006, pp1021-1024, 2006 (in Japanese)
- [2] Peter Albert: Math Class: An Application for Dynamic Tactile Graphics, ICCHP2006, pp1118-1121, 2006