Nigari System - Environment where Students Learn Programming by Creating Animations.

Shinya Cho
Kakehi Laboratory
Waseda University
cho@kake.info.waseda.ac.jp

Abstract

There are too many things to learn when novice students develop programs in well-known languages, such as Java or C. To support learning programming languages, we developed Nigari System[1], a programming environment. The language is simpler than usual ones, and the developing environment enables students make attractive applications, animations or games. As an experiment, we applied Nigari System for a class of programming in Waseda University and reports the result of the experiment.

1 What is Nigari

Nigari is designed to make attractive applications with simple programs.

GUI applications are attractive. Students can make interesting applications with GUI. But libraries are complicated. For example Java AWT package has too many classes to learn. A lot of gadgets are scattered around. Moreover, GUI applications require event-driven programming. It is difficult to understand event mechanisms.

Console-based applications are easy to understand. Not so many gadgets are contained. Tracing program is easy. But most students think console-based applications are boring. These are to perform trivial mathematical calculations.

Nigari is developed as a system that has both attractiveness of products and simpleness in development. Using Nigari, students can display graphics and show animations easily. The library on Nigari is small but enough to perform animations, and has no event mechanism.

2 What 'Nigari' means?

Nigari System is based on Tonyu System[2]. Tonyu System is a development environment for animation programs, but mainly for advanced users. It has more features than Nigari. Some of these features are not necessary for users who has just begun to learning programming. Extracting essential features of Tonyu, we developed Nigari.

Figure 1: Nigari Screenshot

```
while(true) {
  x=x+1; p=2+(x%2);
  if (getKey(38)>0) y=y-1;
  if (getKey(40)>0) y=y+1;
  update();
}
```

Figure 2: Sample Program of Nigari

By the way, Tonyu has some resemblance to Java in language specification. Japanese word *Tonyu* means drink made from soybean juice. We named our system "Tonyu" as "Java" is named by drink made from beans - coffee. *Tonyu* is also raw material of *Tofu*, bean curd. And essence called *Nigari* makes *Tonyu* turn into *Tofu*.

Proceedings of the First Conference on Creating, Connecting and Collaborating through Computing (C5'03)
0-7695-1975-X/03 $17.00 © 2003 IEEE

3 Making Applications in Nigari

To make applications in Nigari, We have following procedures:

- Create a new Page

  A "Page" is a workspace of Nigari. Users can places "Objects" in a Page.

- Create a new Object(s)

  An "Object" is a programmable moving graphic piece. Nigari has a window for creating new Objects(see "New Object Dialog" in Figure 1).

  Objects are placed in the "Page Window". By dragging them the layout of the Page can be edited.

- Write scripts of Objects

  By double-clicking an Object on the screen, Nigari opens the "Code Editor" (see Figure 1). Users write the code in order to define how the Object moves. Each Object can have individual code, so Objects can perform different behaviors.

  Sample code is shown in Figure 2. The variables `x` and `y` represent position of this Object. Just changing values of them, the Object moves. And the variable `p` represents glyph number. Nigari has 60 pre-defined glyphs (see Figure 1).

- Run and debug

  The codes are executed when the user select "Execute" in the menu. Since a thread is assigned to each Object, all Object runs its code simultaneously.

- Submit

  In a programming class, students are usually required to submit programs they wrote. A Page does not consist of a single file. Students are difficult to know which files are necessary to drive their Pages correctly, so Nigari has an auto-submission system, which packs all necessary files into a single file. Students just upload the file for submission.

4 Experiment

To examine how Nigari is useful for students, we used Nigari System in a programming class in Waseda University. The students of the class are freshmen and major computer science. We mainly used Java as a teaching material. In the middle of a term, we introduced Nigari.

Through this experiment, we found major two problems:

- Fundamental troubles

  Students were not used to using computer and had some troubles in basic operations: coping or moving files, extracting archives and executing programs. Currently, Nigari is a Java Application, not a Java Applet. Some of students had stuck in installation phase of Nigari. And when students submit programs they wrote, they have to transfer program files to web server via FTP. This was not so easy to them.

- Confusing

  Students had already studied Java. So they confused Nigari with Java in their specifications. For example, Nigari need not write function/method declarations. Some students wondered why Nigari has no function declaration.

5 Nigari in the future

To solve problems of Nigari that we found, We propose following solutions:

- Nigari Applet

  Since Nigari is Java Application, installation is needed. It is heavy burden to students. By rebuilding Nigari as an Applet, no installation will be needed.

- Nigari files on network.

  Some people had troubles on submission the programs of Nigari. If all files that is needed for Nigari is automatically uploaded to servers, they don't have to do transfer operation.

- Teaching Nigari first

  Once students have studied Java, they don't accept Nigari because they confuse Nigari with Java. We should teach Nigari first and when they used to programming, we should introduce Java in order to write more sophisticated programs. Nigari and Java have some common language specifications, even if we teach some gadgets in Java, students will be able to write programs in Java.

References

[1] Nigari System, 2002.
http://taurus.kake.info.waseda.ac.jp/Nigari/

[2] Tonyu System, 2001.
http://tonyu.kake.info.waseda.ac.jp/