A proposal for teaching programming through the Five-Step Method

Yasuo Uchida, Seigo Matsuno
National Institute of Technology, Ube College
Ube, 755-8555, Japan
E-mail: uchida@ube-k.ac.jp

Takao Ito
Hiroshima University
Higashi-Hiroshima, 739-8527, Japan

Makoto Sakamoto
University of Miyazaki
Miyazaki, 889-2192, Japan

Abstract

We have proposed a new method of advancing from CS Unplugged through the new process of CS Plugged to full-fledged computer programming languages, as a means of deepening understanding in computer programming education. We also have proposed a new Five-Step Method consisting of the following steps: Step 1, A CS Unplugged activity; Step 2, A CS Plugged activity; Step 3, A trace table, Step 4, Preparing pseudocode; and Step 5, Writing Java source code.

Keywords: CS Unplugged, CS Plugged, Programming, Five-Step Method.

1. Introduction

The course we teach has about 40 students per class, aged 17 through 18 years. Since the students' major field of study is Management Information, they need to learn programming [1] techniques. However, the results of the survey described below show that not a few students consider themselves to have insufficient understanding of programming or think that they are not good at programming. We are examining ways to improve this situation. CS Unplugged [2] is a method of teaching information science without using computers, first proposed by Tim Bell of the University of Canterbury in New Zealand. While CS Unplugged is said to be effective in teaching information science [3], there have been concerns that its success or failure may be an effect of the skill and experience of instructors. To address this topic, we implemented CS Unplugged for a group of students of a different age group, from fifth through ninth grades, and looked at their responses. The results showed that after first making sufficient preparations the method could generate results without necessarily depending on the skill or experience of instructors. Other research underway in Japan includes a study on use of teaching aids in learning about algorithms through CS Unplugged [4] and a study on

Published by Atlantis Press

learning the fundamentals of computer programming through a programming learning environment for beginners [5]. A study by Y. Feaster et al [6] concerns practice in teaching high-school students, and it has been reported to have had some, albeit limited, success.
However, at present almost no research has been conducted on advancement from CS Unplugged to full-fledged programming languages. Accordingly, we propose a new method of advancing from CS Unplugged to full-fledged programming. The proposed method begins with conducting a CS Unplugged activity, and then continues on to writing a program on the same theme and further to its abstraction in Java. Accordingly, we propose advancing from CS Unplugged to full-fledged programming through a new Five-Step Method. The proposed method consists of the following steps: Step 1, A CS Unplugged activity; Step 2, A CS Plugged activity; Step 3, A trace table, Step 4, Preparing pseudocode; and Step 5, Writing Java source code.

2. Background of this study

Every year we conduct a survey following our first-term course of two 90-minute sessions per week held over 30 weeks. Here we will look at numbers of students who answered “yes” or “no” to this survey’s question, “Do you think you largely understand Java?”

As seen in Fig. 1, the survey’s results show that not a few students consider themselves to have insufficient understanding of programming or think that they are not good at programming. As such, there is a need to improve this situation.

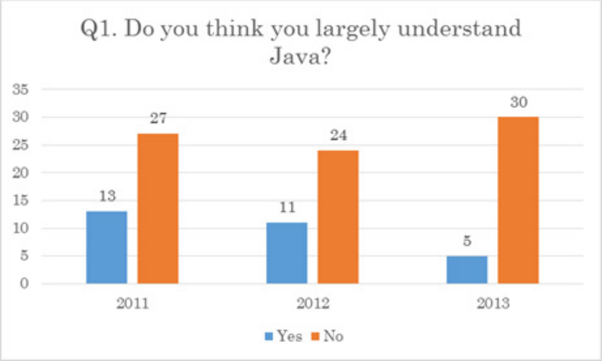


Fig. 1. Survey results on understanding of Java.

3. CS Unplugged in practice

For about 50 minutes on August 2, 2014 we used the CS Unplugged method for approximately 30 students from fifth through ninth grades as part of a summer-vacation junior science course. The activity we implemented was CS Unplugged’s Image Representation activity. Fig. 2 shows examples of students’ work.

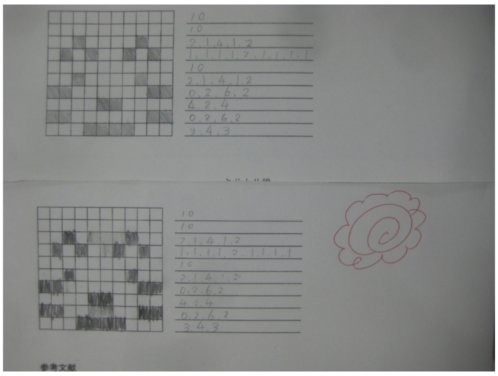


Fig. 2. Examples of students’ work

4. The Five-Step Method

In this paper, we use the name CS Plugged to refer to implementing a CS Unplugged activity through a computer program. The goal is to advance to computer programming through using a computer program to conduct the work done by human beings in a CS Unplugged activity.

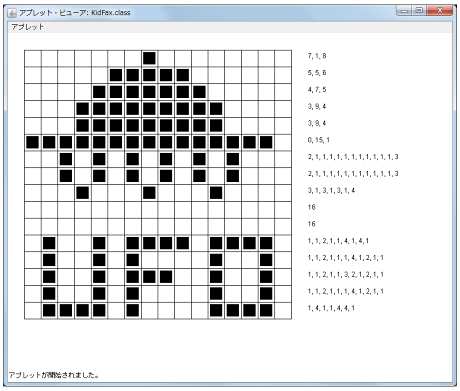
Accordingly, we propose advancing from CS Unplugged to full-fledged programming through a Five-Step Method. The proposed method consists of the following steps: Step 1, A CS Unplugged activity; Step 2, A CS Plugged activity; Step 3, A trace table, Step 4, Preparing pseudocode; and Step 5, Writing Java source code.

4.1. *Step 1: CS Unplugged*

First, students conduct the CS Unplugged activity as described in Section 3 above.

4.2. *Step 2: CS Plugged*

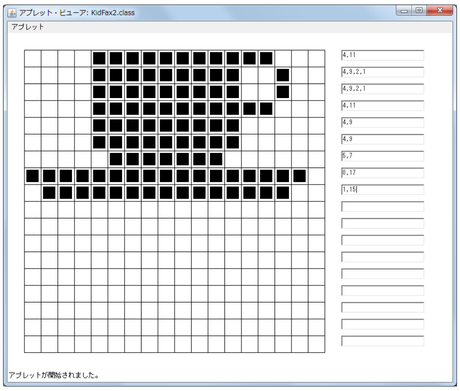
Here we will look at the example of using a computer program to implement the CS Unplugged activity Image Representation. This is conducted through two activities. The first, converting the image to code, is represented in Fig. 3. When students click on squares in the grid at left to draw a picture, the image is converted instantly to code as displayed at right.


アプレット・ビューア: KidFax.class
アプレット
アプレットが開始されました。

```
7, 1, 8
5, 5, 6
4, 7, 5
3, 9, 4
3, 9, 4
0, 15, 1
2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3
2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3
3, 1, 3, 1, 3, 1, 4
16
16
1, 1, 2, 1, 1, 4, 1, 4, 1
1, 1, 2, 1, 1, 1, 4, 1, 2, 1, 1
1, 1, 2, 1, 1, 3, 2, 1, 2, 1, 1
1, 1, 2, 1, 1, 1, 4, 1, 2, 1, 1
1, 4, 1, 1, 4, 4, 1
```

Fig. 3. Applet for converting image to code.

The other activity is the reverse of this process, with students reproducing a picture from code (Fig. 4). They enter code to the text boxes at right in a format such as “7,1,8” and press the Enter key to display the resulting image instantly in the grid at left.


アプレット・ビューア: KidFax2.class
アプレット
アプレットが開始されました。

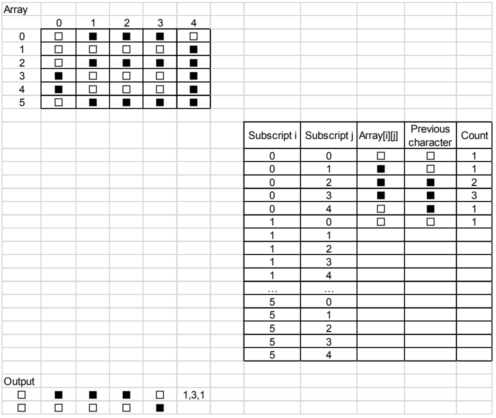
```
4,11
4,9,2,1
4,9,2,1
4,11
4,9
4,9
5,7
0,17
1,15
```

Fig. 4. Applet for reproducing an image

4.3. *Step 3: Trace table*

The task conducted in the CS Plugged activity is analyzed and diagrammed as a trace table [7]. This is designed for entry of not just the execution process but also the data areas of arrays (including initial values) and output areas.

Fig. 5 shows an example of a trace table prepared using a spreadsheet software, although the actual trace table may be drawn by hand.



Array

	0	1	2	3	4
0	□	■	■	■	□
1	□	□	□	□	■
2	□	■	■	■	■
3	■	□	□	□	■
4	■	□	□	□	■
5	□	■	■	■	■

Subscript i	Subscript j	Array[i][j]	Previous character	Count
0	0	□	□	1
0	1	■	□	1
0	2	■	■	2
0	3	■	■	3
0	4	□	■	1
1	0	□	□	1
1	1			
1	2			
1	3			
1	4			
…	…			
5	0			
5	1			
5	2			
5	3			
5	4			

Output

□	■	■	■	□	1,3,1
□	□	□	□	■	

Fig. 5. Sample Trace Table (in process)

4.4. *Step 4: Preparing pseudocode*

In this step, the three elements of sequence, decision, and repetition through preparation of the trace table in the preceding step are extracted and represented in pseudo-language. In Japan, the national Information Technology Engineer Examinations employ pseudo-language [8]. A pseudo-language simulator is a type of software that makes such pseudo-language executable. In this paper, we used the freeware pseudo-language simulator SARA [9]. Fig. 6 shows the code written in this step and its execution.

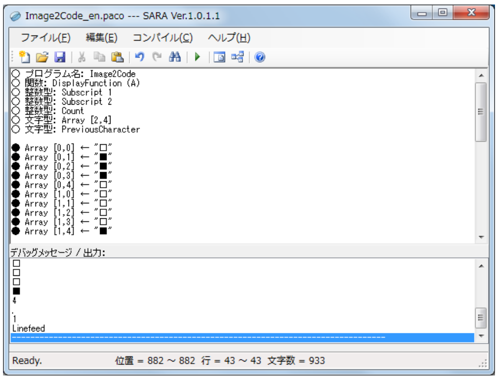


Image2Code_en.paco --- SARA Ver.1.0.1.1
ファイル(F)
編集(E)
コンパイル(C)
ヘルプ(H)
デバッグメッセージ / 出力:
Ready.

```
○ プログラム名: Image2Code
○ 関数: DisplayFunction (A)
○ 整数型: Subscript 1
○ 整数型: Subscript 2
○ 整数型: Count
○ 文字型: Array [2,4]
○ 文字型: PreviousCharacter

● Array [0,0] ← "□"
● Array [0,1] ← "■"
● Array [0,2] ← "■"
● Array [0,3] ← "■"
● Array [0,4] ← "□"
● Array [1,0] ← "□"
● Array [1,1] ← "□"
● Array [1,2] ← "□"
● Array [1,3] ← "□"
● Array [1,4] ← "■"
```

```
□
□
□
■
4
,
1
Linefeed
--------------------------------------------------------------------------------
```

```
位置 = 882 ～ 882  行 = 43 ～ 43  文字数 = 933
```

Fig. 6. Sample use of pseudo-language simulator

4.5. *Step 5: Writing Java source code*

In this step students create a program by converting the pseudo-language from the previous step to Java source code. The subsequent debugging process is handled by going back and examining each previous step depending on the content of the error messages.

Published by Atlantis Press

- Java source code

```
public class Image2Code {
   public static void main(String args[]) {
      char[][] image = {{'□', '■', '■', '■', '□'},
                        {'□', '□', '□', '□', '■'},
                        {'□', '■', '■', '■', '■'},
                        {'■', '□', '□', '□', '■'},
                        {'■', '□', '□', '□', '■'},
                       {'□', '■', '■', '■', '■'}
                      };
         // Iterate only number of lines in an array
         for(int i = 0; i < image.length; i++) {
            char previous = '□';
            int count = 0;
         // Display the image in pixels
         for(int j = 0; j < image[0].length; j++) {
            System.out.print(image[i][j]);
         }
         System.out.print(" ");
         // Count number of adjoining pixels of same color and
display in numerical form (code)
         for(int j = 0; j < image[0].length; j++) {
            if (image[i][j] == previous) {
               count++;
            } else {
               previous = image[i][j];
               System.out.print(count + ", ");
              count = 1;
            }
         }
         System.out.println(count);
      }
   }
}
```

- Results of running the above program

```
□■■■□ 1, 3, 1
□□□□■ 4, 1
□■■■■ 1, 4
■□□□■ 0, 1, 3, 1
■□□□■ 0, 1, 3, 1
□■■■■ 1, 4
```

5. Conclusion

We have proposed a new method of advancing from CS Unplugged through the new process of CS Plugged to full-fledged computer programming languages, as a means of deepening understanding in computer programming education. We also have proposed a new Five-Step Method consisting of the following steps: Step 1, A CS Unplugged activity; Step 2, A CS Plugged activity; Step 3, A trace table, Step 4, Preparing pseudocode; and Step 5, Writing Java source code. Topics for the future are those of putting together detailed procedures for abstraction of pseudocode from trace table as well as continually implementing the proposed method and measuring its results.

References

1. D. Everitt, Introducing programming skills in education, https://www.academia.edu/1478617/Introducing_programming_skills_in_education.
2. The Unplugged Community: Computer Science Unplugged, http://csunplugged.org/.
3. Y. Idosaka, Y. Kuno, and S. Kanemune, Attempt and the Practice of Class Method Improvement Based on "Computer Science Unplugged", *Journal of the Japan Society of Technology Education*, vol.53, no.2 (2011), pp. 115-123 (in Japanese).
4. H. Manabe, S. Kanemune, and M. Namiki, Effects of Teaching Tools in CSU Algorithm Education, *IPSJ Journal*, vol.54, no.1 (2013), pp. 14-23 (in Japanese).
5. T. Nishida, A. Harada, R. Nakamura, Y. Miyamoto, and T. Matsuura, Implementation and Evaluation of PEN: The Programming Environment for Novices, *IPSJ Journal*, vol.48, no.8 (2007), pp. 2736-2747 (in Japanese).
6. Y. Feastery, L. Segarsz, S.K. Wahbay, J.O. Hallstrom, Teaching CS Unplugged in the High School (with Limited Success). Proceedings of the 16th annual joint conference on Innovation and technology in computer science education, pp. 248--252 (2011)
7. Cambridge International Examinations, SYLLABUS Cambridge O Level Computer Studies 7010 For examination in June and November 2014, *University of Cambridge International Examinations*, http://www.cie.org.uk/ images/90258-2014-syllabus.pdf.
8. Information-technology Promotion Agency Japan, Kyotsu ni shiyo sareru gijigengo no kijutsukeishiki ("Symbolic conventions of commonly used pseudo-language"), https://www.jitec.ipa.go.jp/1_13download/ gijigengo_keisiki.pdf (in Japanese).
9. S. Mimura, Gijigengo shimyureta SARA ("SARA pseudo-language simulator"), http://mimumimu.net/ software/#sara (in Japanese).