

Learning Based On Computer Science Unplugged in Computer Science Education: Design, Development, and Assessment

Eiko Takaoka, Yoshiyuki Fukushima, Koichiro Hirose, Tadashi Hasegawa

Abstract—Although, all high school students in Japan are required to learn informatics, many of them do not learn this topic sufficiently. In response to this situation, we propose a support package for high school informatics classes. To examine what students learned and if they sufficiently understood the context of the lessons, a questionnaire survey was distributed to 186 students. We analyzed the results of the questionnaire and determined the weakest units, which were “basic computer configuration” and “memory and secondary storage”. We then developed a package for teaching these units. We propose that our package be applied in high school classrooms.

Keywords—Computer Science Unplugged, computer science outreach, high school curriculum, experimental evaluation.

I. INTRODUCTION

IN 2003, information studies was a required course at most high schools in Japan. Starting in 2013, taking a course on information studies is now a requirement for all newly admitted high school students. Thus, teachers are now attempting to develop various teaching methods. For example, a teaching method that uses “Dolittle”, a software package in which students create programs in Japanese, is suitable for information education [1]. Furthermore, the following are some examples of information education: using music as a teaching material, and using the teachers’ own teaching materials. Consequently, the number of students that are interested in studying a programming language, which tends to be considered difficult, is increasing. However, some people point out that “the learning environment is unsuitable because it is too pleasant” [2]. In other words, they feel that students should not only enjoy these materials, they should also be gaining a deeper understanding of computer science while using the abovementioned kinds of audio/visual materials. Therefore, some high school teachers need a software package as a teaching material for teaching information studies.

Recently, some schools provide computer science education from the elementary to college level by using Computer Science Unplugged, which is a set of 20 activities that was

developed by Tim Bell and his colleagues at the University of Canterbury in New Zealand [3]. This method teaches computer science through engaging games and puzzles that use cards, string, crayons, and physical movement, but not computers [3].

With the above conditions in mind, we constructed an educational method based on Computer Science Unplugged that we believe will be useful for students who wish to obtain the amount of information knowledge that is required by society today, and to hopefully interest them in information technology in information education. Our package was built to be not only pleasant to use, but also to focus largely on the fields that are important and difficult for students to understand, thus encouraging them to learn much more.

In the first phase, we investigated and specified the field that students most want to learn, but consider to be difficult to understand. This field was “the composition of a computer and a processing unit”. Next, we proposed a role-playing method as a computer education package, in which students do not use a computer and practice reports are completed.

II. RELATED WORKS

In addition to the above-mentioned Computer Science Unplugged, there have been various other attempts to influence student views and attitudes towards computer science through the use of games and pedagogical software tools.

These include drag-and-drop learning environments like Alice [4] and Scratch [5], a visual, block-based programming language designed to facilitate media manipulation for novice programmers, and thus teach them how to eliminate syntax errors. This proved to be an attractive way to introduce programming concepts to students.

There are many case studies showing that Alice and Scratch had significant impacts on the educational effects of learning computer science for high school students [6]–[8].

On the other hand, Feaster et al. [9] reported that a high school course based upon the Computer Science Unplugged program raised questions related to determining how and how not such programs should be used in classrooms.

To address these questions, in particular, we will evaluate the activities in Computer Science Unplugged with respect to determining the fields that are both important and difficult for students to understand.

III. DIFFICULT FIELD TO UNDERSTAND

We began by distributing questionnaires and conducting

Eiko Takaoka is with Faculty of Science and Technology, Sophia University, Kioi-cho7-1, Chiyoda-ku, Tokyo, Japan (e-mail: m-g-eiko@sophia.ac.jp).

Yoshiyuki Fukushima is with Graduate School of Science and Technology, Sophia University.

Koichiro Hirose is with Tokyo Metropolitan Tama High School of Science and Technology, Honcho6-8-9, Koganei, Tokyo, Japan.

Tadashi Hasegawa is with Faculty of Bioscience and Applied Chemistry, Hasei University, 3-7-2, Kajino-chou, Koganei, Tokyo, Japan.

interviews among the students at “T” high school, which has a department of information technology [10]. Specifically, we investigated their course entitled “Foundation for information technology”. The objectives of this course aim at teaching students the fundamental knowledge and technology of information technology, giving them the capability to utilize information and information devices, showing the progress of computerization, and exploring the meaning of information in society. The questionnaire was distributed in accordance with the items in the textbook “Seisenjohogijutukiso” published by Jikkyo Shuppan, which is currently used at “T” high school [11].

Students were required to rate their own comprehension level for each of the following learning units: computer configuration and processing units, logic circuits, binary systems and decimal systems, computer security, and copyright and portrait rights. A five-point rating scale was used (1: Very poor, 2: Poor, 3: Average, 4: Good, 5: Very good).

Fig. 1 shows the results of the questionnaire survey. The results indicate that the most difficult fields for the students to understand include computer configuration and processing units, whereas the most easily understood fields include copyright and portrait rights. Therefore, we developed learning materials based on Computer Science Unplugged to facilitate the learning of computer configuration and processing units.

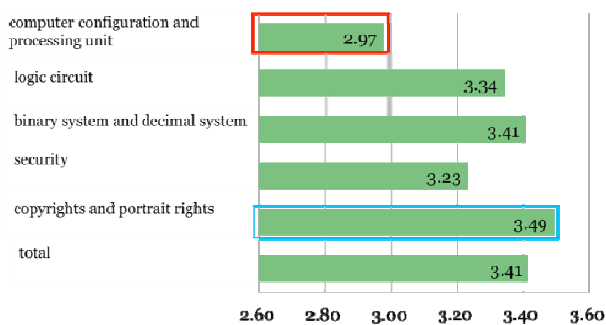


Fig. 1 Comprehension Level

IV. METHODS

We developed a role-playing package to assist in learning of the configuration of a computer, main memory, and secondary storage device. The package consists of a summary, materials list, introduction, a scenario, a program used for the specific scenario, and a summarizing discussion.

A. Role-Playing: Basic Computer Configuration

1. Summary

The circuitry in a computer that controls the manipulation of data is called the central processing unit, or CPU. For the purpose of transferring bit patterns, the CPU and main memory are connected by a collection of wires called a bus. Through this bus, the CPU reads data from the main memory. The computer performs several operations (input, output, storing, and controlling) to carry out each task. How can a computer perform such operations and manipulate data? Let us think

about that via role-playing.

2. Materials

- A1 size paper for the “computer’s main memory”
- A4 size paper for the program and variables (shown in the Fig. 2)
- Pen

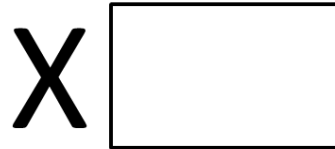


Fig. 2 Space for variables

3. Introduction

Choose five pupils to play the role of each unit, as shown below.

- MC (program counter): instructs each pupil how to act.
- Control Unit: gives instructions to each unit, reads data from the main memory, and writes data to the main memory.
- Input Device: retrieves data from the peripheral devices and writes them to the main memory.
- Arithmetic/Logic Unit: carries out the operations.
- Output Device: states the values of the variables.

4. Program (written in the paper in advance)

- Reserve storage for variables X and Y.
- Input value into the X variable
- Input value into the Y variable.
- Execute $X+Y$.
- Output the result of $X+Y$.

5. Scenario

MC: “Let’s start the execution”

(MC indicates the first line of the program.)

(Control Unit sticks the A4 papers, which have the names and areas of the variables written on them, to the A1 papers.)

Control Unit: “Input Device, please begin your preparation.”

(MC indicates the second line of the program and shows the value “4” to the Input Device.)

(Input Device confirms the value of “4” and writes the value into the area for the X variable with a pen.)

(MC indicates the third line of the program and shows the value “3” to the Input Device.)

(Input Device confirms the value “3” and writes the value into the area for the Y variable with a pen.)

(MC indicates the fourth line of the program.)

Control Unit: “ALC, please execute the addition.”

(ALC retrieves the data (arguments) for the addition, calculates $4+3$, and writes the result onto the paper.)

(MC indicates the fifth line of the program.)

Control Unit: “Output Device, say the result of the addition.”

Output Device reads the result from the paper.

Output Device: “The result is 7.”

6. Discussion

A CPU consists of two parts: the arithmetic/logic unit, which contains the circuitry that performs the operations on the data (such as additions and subtractions), and the control unit, which contains the circuitry for coordinating the activities of the machine.

A CPU contains a program counter, which indicates where the CPU executes. The role of the program counter in the scenario is played by MC.

A computer follows a program stored in its memory by copying the instructions from the memory into the control unit. The control unit controls the input device, arithmetic/logic unit (ALU), and the output device.

B. Role-Playing: Main memory and the Secondary Storage

1. Summary

For the purpose of transferring bit patterns, the main memory and the secondary storage device, such as hard disks, CDs, DVDs, flash drives, and memory cards, are connected by a collection of wires called a bus. What occurs when you click the "icon"? Programs are stored in the hard disk. When the program is called, the control unit retrieves the program from the hard disk and reads it into the main memory. For executing the program, the execution environment is also reserved into the memory. Main memory and the secondary storage

2. Materials

- AA1 size paper for the "computer's main memory"
- A1 size paper for "additional memory"
- A4 size paper for the programs for the function of studying math and English, and their related math textbook, math notebook, English textbook, English notebook, and a dictionary.
- A bag (or something similar) to represent the hard disk

3. Introduction

Prepare the materials and choose one pupil to play the role of the control unit, as shown below.

- MC: instructs how each pupil should act.
- Control Unit: transfers the program to the environment for the operations between the main memory and secondary storage device.
- Main memory (A1 paper): Programs are stored here.
- Secondary storage device (bag): Where the programs are stored.

4. Program (Written in the Paper in Advance)

- Studying the math program
 - Read the math textbook from the secondary storage device into the main memory.
 - Read the math notebook from the secondary storage device into the main memory.
- Studying the English program
 - Read the English textbook from the secondary storage device into the main memory.
 - Read the English notebook from the secondary storage device into the main memory.
 - Read the English dictionary from the secondary

storage device into the main memory.

5. Scenario

MC: "Suppose that there is an icon that represents the action of studying math. When you click on the icon, you have the option to study math. Control Unit, please read the math program from the hard disk and into the main memory."

(Control Unit takes the paper with "math program" written on it from the bag and sticks it to the A1 paper.)

MC: "Control Unit, when the program executes, read the math textbook and the math notebook into the memory."

(Control Unit takes the math textbook and notebook from the bag and sticks it to the A1 paper.)

MC: "Now you also want to study English in addition to studying math. There is an icon that represents the study of English. When you click on the icon, you have the option to study English. Control Unit, please read the English program from the hard disk and into the main memory."

(Control Unit takes the paper with "English program" written on it from the bag and sticks it to the A1 paper.)

MC: "Control Unit, when the program executes, read the English textbook, English notebook, and dictionary into the memory."

(Control Unit takes the English textbook, notebook, and dictionary from the bag and tries to stick them all to the A1 paper. However, there is not enough space for all three items and the dictionary cannot be added.)

MC: "Notice that the memory is exhausted and the computer has been freezing." Fig. 3 shows that situation.

(MC sticks the additional memory beside the main memory.)

MC: "If you add more memory, the dictionary is able to be stored in the memory."

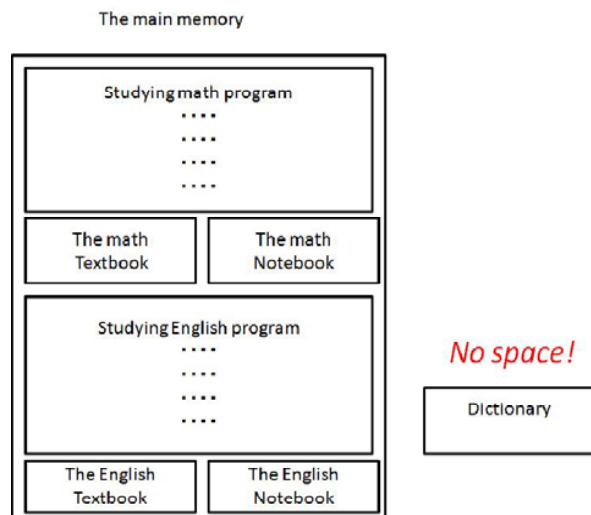


Fig. 3 The memory is exhausted and the computer has been freezing

6. Discussion

By going through this scenario, students learn that the main memory is used for the program execution environment. When the two programs are executed concurrently, the memory is exhausted and the computer freezes.

Consider that Excel and Word are the programs used to study

math and English, respectively. If both programs are executed concurrently, there is a possibility that your computer will freeze. It is important not to execute numerous programs concurrently and to save your files frequently.

V. PRELIMINARY EXPERIMENT FOR HIGH SCHOOL STUDENTS

We provided lectures by using the educational package we developed to 36 students at "T" high school on January 16, 2013. The class was divided into two groups by their names in alphabetical order. In the first group (experimental group), the representatives performed the procedures, and the other students observed and learned the procedures. In the second group (control group), the students were placed in a traditional class that did not use role-playing. After the procedures were taught, the students took an examination.

A. Examination

After role-playing, the lecturer explained the relationship between the role-playing game and the actual computer configuration. The students then took an examination. The examination consisted of four questions. Questions 1, 2, and 3 pertain to the basic configuration of a computer. For Question 1, students are required to understand the connection between the names of each of the devices (input device, output device, main memory, control unit, and arithmetic/logic unit) and their respective roles (5 points out of 30). Question 2 is a fill-in-the-blank question about the name of one of the devices (5 points out of 30). For Question 3, the students are required to understand the relationship between the names of the devices; for example, a mouse is the input device, a speaker is the output device, and so on (14 points out of 30). Question 4 pertains to the memory and secondary storage device (6 points out of 30).

B. Role-Playing

In the experimental group, the representatives performed the procedures of the scenario. Each representative performed a procedure about the basic configuration of a computer (arithmetic/logic unit, input device, output device, and control unit). The main memory was provided as a large piece of paper that was placed on a blackboard, as shown in Fig. 4.



Fig. 4 Representatives identifying the basic configuration of a computer

C. Results

Table I shows the average scores of the examination for the first group (role-playing lesson) and second group (traditional lesson). For the total score and the score for Questions 1 and 3, the average score for the first group was higher than that of the second group. No significant difference was observed between the groups for Question 1. For Questions 2 and 4, the average score of the first group was higher than that of the second group. A significant difference was observed between both questions ($t(28) = 1.76, p < 0.1$).

According to these results, the performance of the experimental group was determined to be higher than that of the control group. Thus, it can be said that learning based on Computer Science Unplugged increases the ability of a student to learn effectively.

TABLE I
AVERAGE SCORES OF THE EXAMINATION (*: $p < 0.1$)

	Q1	Q2	Q3	Q4
First group (experimental group)	4.8	4.4*	13.4	5.3*
Second group (control group)	4.8	4.1	13.7	4.9

VI. PRACTICE FOR COLLEGE STUDENTS

A. "I" University

We provided lectures created using the educational package we developed to the "Computers and Human Interaction" course at "I" University. The numbers of students in these courses were 64 and 59 in 2012 and 2013, respectively. In 2012, the representatives performed while the other students observed them. In 2013, the students took a traditional style lesson that did not use role-playing.

All students took an examination after learning. Table II shows the average scores of the examination in 2012 and 2013. As can be seen in this table, the average score of 2012 was higher than that of 2013. A significant difference was observed between both questions ($t(106) = 2.19, p < 0.05$). Therefore, as mentioned above, the validity of the developed package was proved.

TABLE II
AVERAGE SCORES OF THE EXAMINATION (*: $p < 0.05$)

	Average Scores
2012	27.0*
2013	25.5

B. "S" University

We provided lectures created using our educational package to "Information and human interaction" at "S" University. The number of students in this course was 63. At the beginning of the term, the students took a preliminary examination. This test differs from the procedure for "T" high school students and consisted of questions in which they are required to describe the name, role and examples of the configuration of a computer (26 points). The class was divided into three 21-person groups based on their names in alphabetical order in an attempt to ensure that the average of the examination results would be

equal. The students of the first group studied by themselves using the lesson materials regarding computer configuration. In the second group, representatives performed role-play while other students in that group observed. In the third group all the members were divided into smaller groups of three or four students, and each student was given a role to perform. The students took the same examination after learning as they took beforehand.

Table III shows the average scores of the examination for the first group (self study), the second group (role-playing by representatives) and the third group (role-playing by all students). There were no significant differences in the results of the post examination for the three methods. However, the score growth before and behind the study for every group was in the order of the third group (role-playing by all students), the second group (role-playing by the representative), and the first group (self study). Therefore for college students, it is thought that role-playing by all students is most suitable method.

TABLE III
AVERAGE SCORES OF THE EXAMINATION

	Preliminary examination	Post examination
First group (self study)	3.9	17.0
Second group (role-playing by the representative)	3.8	17.8
Third group (role-playing by all students)	3.8	18.5

VII. PRACTICE FOR HIGH SCHOOL STUDENTS

Next, we consider which method is more effective, the method in which representatives perform the procedures while other students watch or the method in which all the students perform the procedures.

To compare these methods, we provided lectures using our developed educational package to 216 students at "T" high school on October 22, 23, and 25, 2013 [12]. Up until these lectures, the participating students had never studied "the composition of a computer and processing unit". Six classes were held, and each class consisted of 36 students. We divided each class into two groups by their names in alphabetical order. The first group consisted of representatives that performed the procedures while the other students observed and learned the procedures. For the second group, all the members were divided into smaller groups of three or four students, and each student performed the procedures. After the procedures were performed, the students took the same examination as described for the preliminary practice test. Feedback was also obtained from the students.

A. Role-playing: Basic Computer Configuration

During the preliminary practice, only the representatives performed the procedures. For this case, however, the students in the second group were divided into smaller groups of three or four and each member performed the procedures related to the basic configuration of a computer (arithmetic/logic unit, input device, output device, and control unit). Fig. 5 shows the scenario in which the representatives performed the procedures

related to the basic configuration of a computer. Fig. 6 shows the scenario in which all of the students engaged in role-playing.

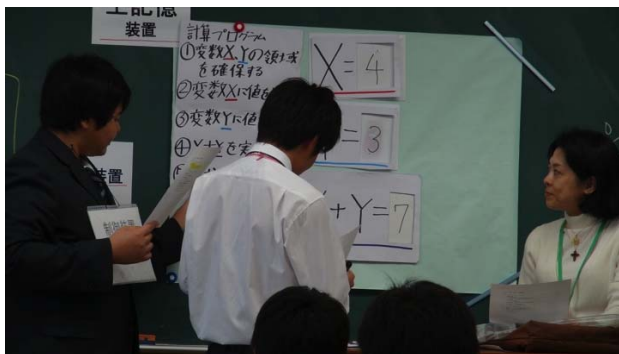


Fig. 5 Representatives role-playing procedures of the basic configuration of a computer

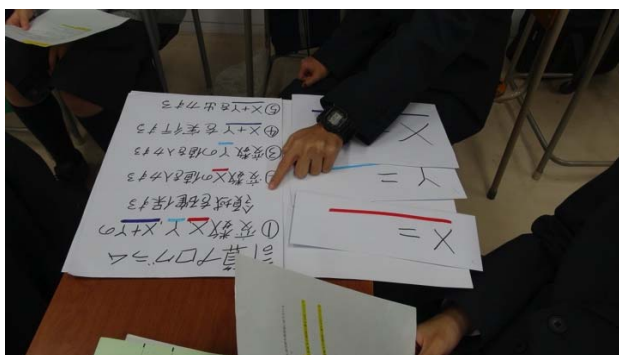


Fig. 6 All of the students role-playing the procedures of the basic configuration of a computer

B. Role-playing: Memory and Secondary Storage Device

After role-playing the configuration of a computer, the students performed the procedures for the memory and secondary storage device of a computer. Fig. 7 shows the scenario in which the representatives performed the procedures for the memory and auxiliary storage device. Fig. 8 shows the scenario in which all of the students engaged in role-playing.



Fig. 7 Representatives role-playing the procedures of the memory and auxiliary storage device



Fig. 8 All of the students role-playing the procedures of the memory and auxiliary storage device

C. Results

According to the feedback from the students, there were only a few negative opinions. One student had many positive comments. For example, the student commented, "I now understand the role of the basic configuration and internal hardware of a computer after role-playing the procedures." One high school teacher said, "It was impressive to see that the students that are typically quiet were happy and active while performing the procedures."

Table IV shows the average scores of the examination for the first group (role-playing by the representatives) and second group (role-playing by all students). For the total score and Questions 1 and 3, the average score of the first group was higher than that of the second group. A significant difference was observed between the scores for Question 1 ($t(198) = 1.82$, $p < .1$).

TABLE IV
AVERAGE SCORES OF THE EXAMINATION (*: $P < 0.05$)

	Q1	Q2	Q3	Q4	Total
First group	4.54*	4.51	13.5	5.49	28.1
Second group	4.40	4.52	13.3	5.51	27.8

We compared the average scores of the four questions and total between the first and second groups for the six classes. However, there was no significant difference between the two methods. Therefore, there is no significant difference between the first group (role-playing by representatives) and second group (role-playing by all students).

VIII. CONCLUSION

We investigated the learning field that was most difficult for students to understand during a lesson for a course titled "Foundation of information technology". From the results of our questionnaire survey, we determined that this field was "the composition of a computer and processing unit". Therefore, we propose a learning method based on Computer Science Unplugged. Our method uses role-playing to teach the concepts of this unit and can be applicable to high school classrooms.

The activities we propose can also be applied to university subjects. We already have provided our activities on the Computer Science Unplugged website where anyone can access our materials and download the documents described in

Section III:

(<http://csunplugged.org/activities-under-development>).

In the future, we will apply our activities to universities or other high schools.

REFERENCES

- [1] Programming Language "Dolittle", <http://dolittle.eplang.jp/> (Access 30 January 2014).
- [2] T. Tatsumi, "Why don't you take your idea into lectures?" IPSJ Magazine. Information Processing Society of Japan, 2011, Vol. 52, No. 6, pp. 735.
- [3] Computer Science Unplugged. <http://csunplugged.org/> (Access 30 January 2014).
- [4] Alice.org, <http://www.alice.org/index.php>, (Access 30 January 2014).
- [5] Scratch -Imagine, Program, Share, (Access 30 January 2014)
- [6] S. Cooper, W. Dann, and R. Pausch, "Teaching Objects-first in Introductory Computer Science", 34th SIGCSE Technical Symposium, 2003, pp. 191-195.
- [7] J. Adams, "Alice, Middle Schoolers, and the Imaginary Worlds Camps", 38st SIGCSE Technical Symposium, 2007, pp. 307-311.
- [8] J. Maloney, K. Peppler, Y. Kafai, M. Resnick, and N. Rusk, "Programming by Choice: Urban Youth Learning Programming with Scratch", 39th SIGCSE Technical Symposium, 2008, pp. 367-371.
- [9] Y. Feaster, L. Segars, S. K. Wahba, and J. O. Hallstrom, "Teaching CS unplugged in the High School (with limited success)", 16th Annual Joint Conf. on Innovation and Technology in Computer Science Education, ITICSE 2011, pp. 248-252.
- [10] Y. Fukushima, K. Hirose, T. Hasegawa, and E. Takaoka, "A study on information education in a high school: Toward Development of the package for learning computer science", SSS 2012 (20-22 August 2012), Information Processing Society of Japan, Vol. 2012, No. 4, pp. 169-175, in Japanese.
- [11] M. Iwamoto, H. Iwamoto, et al., 2011, Seisenjohogijutukiso, JikyoShuppan Co., Ltd.
- [12] Y. Fukushima, K. Hirose, T. Hasegawa, and E. Takaoka, "A proposal of a package for Computer Science Unplugged and practice at a high school". Vol. 2013-CE-122, pp. 1-6, Information Processing Society of Japan, in Japanese.