Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 49-19 (2016) 514–518

Tangible gimmick for programming education using RFID systems

Tatsuo. Motoyoshi * **Naoki. Tetsumura** * **Hiroyuki. Masuta** *
Ken'ichi. Koyanagi * **Toru. Oshima** * **Hiroshi. Kawakami** **

* *Toyama Prefectural University, Imizu, Japan (e-mail: motoyosh@pu-toyama.ac.jp)*
** *Unit of Design, Kyoto University, Kyoto, Japan, (e-mail: kawakami@design.kyoto-u.ac.jp)*

Abstract: We developed programming tools which has a tangible user interface. One is P-CUBE and another is Pro-Tan. People who are inexperienced in operation a personal computer including visual impairments can learn fundamental concepts of programming by using these tools, since users are able to make a program only by positioning cards or blocks on a board without operating PC's interfaces. These tools utilize radio frequency identification (RFID) tags alone for detecting the position of cards or blocks, and thus require no precision equipment such as microcomputers. These are robust for rough operation of children by such a simple mechanism. Furthermore, since programming blocks of P-CUBE has tactile information for discriminating block types, visual impairments are able to make a program without supports. In this paper, we report on the system configuration of P-CUBE and Pro-Tan and effectiveness of thise tangible programming tools. In addition, we evaluate advantages of using thus tangible programming tools, and analyze the characteristic of these tools as a gimmick of programming education.

Keywords: Programming Education, Gimmick, Tangible, Visually Impairements, RFID

1. INTRODUCTION

A number of people studying programming from elementary school has been increasing because of the increasingly information-intensive society. We think there is a need for programming education tool, which is easy to operation for beginners who are inexperienced in personal computer operation including visually impaired. A number of studies on programming education tools for younger individuals have been conducted(1; 2; 3; 4; 5; 6). Although, the complicated operation should be learned before using such programming tools, these are often an effective method only for people who are accustomed to using a personal computer (PC) and learners must depend on visual information.

We are developing tangible(7) programming tool which is a education tool for beginners and visually impaired(8). One is P-CUBE and another is Pro-Tan.

P-CUBE and Pro-Tan are intended to teach fundamental programming concept(10; 11) in beginner-level by simple operation. When using these tools, users are able to make a program only by positioning cards or blocks on a mat without operating a programming software. These tools are intended to be a trigger as it is called "shikake" for the programming learning which are proposed in shikakeology(12). These systems utilize radio frequency identification (RFID) tags alone for detecting the position of blocks and cards, and thus require no precision equipment such as microcomputers. Furtheremore, we design interfaces of these tools intended for easy understanding how to use blicks or cards. These tools are considered as gimmicks for programming education.

In this paper, we report on the system configuration of P-CUBE and Pro-Tan and how to use these tools. Then, we describe characteristics these tools as a gimmick.

2. P-CUBE

2.1 System configuration

P-CUBE consists of a program mat, programming blocks, and a PC (8). Users are able to create four types programs (sequential OPEN, sequential LOOP, conditional branch - one sensor, and two sensors) for mobile robot using only by positioning programming blocks on the program mat. Programming blocks has tactile information like motion directions to present type of blocks to users. Fig.1 shows the concept of P-CUBE.

2.2 Programming block

Fig.2 shows all types of programming blocks. There are four types of programming blocks motion, timer, IF, and LOOP blocks. Programming blocks correspond to the mobile robot movements, loop instruction and conditional branch instruction as program elements.

⋆ This work was supported by JSPS KAKENHI Grant Number 26870483

2405-8963 © 2016, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2016.10.608

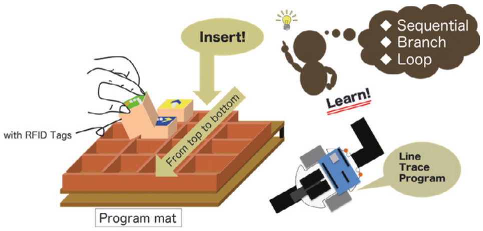

Fig. 1. P-CUBE concept

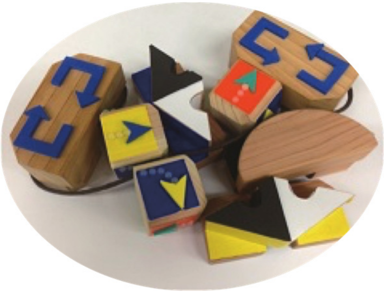

Fig. 2. Programming blocks of P-CUBE

Motion block Motion blocks have four RFID tags on each face and instruct the mobile robot to move forward, back, or rotate depending on the face displayed to the sensors. Since each side of a motion block corresponds to a mobile robot motion, users can control movements of the mobile robot by changing the displayed side of the motion block. The blocks are made from Japanese cedar and weigh approximately 30 grams.

Timer block Timer blocks set the movement duration of the robot on a scale of one to four. Users have to set timer blocks next to motion blocks. The timer blocks are also made from cork and weigh approximately 16 grams.

IF block IF blocks correspond to the conditional branch functions depending on the information from two infrared (IR) sensors mounted in the mobile robot. Blocks consists of a begin block and an end block. Using IF blocks, the user can create a line trace program. Tactile information is incorporated into begin blocks by equipping them with silicone rubber dots, ethylene-vinyl acetate (EVA) sheets and drawing paper. These materials are also used on the course mat itself. The input from the IR sensor is used to distinguish between the black and white parts on the course mat. The blocks are made from Japanese cedar and weigh approximately 60 grams.

LOOP block LOOP blocks correspond to while loop functions. Mobile robot repeats the movements of blocks positioned between a pair of LOOP blocks. LOOP blocks are made from Japanese zelkova wood and weigh approximately 205 grams.

2.3 Programming operation

Users can complete a program by positioning programming blocks on the program mat adequately. Operation sequence of P-CUBE is as follows.

(1) Select an adequate programming blocks
(2) Position programming blocks on the program mat
(3) Slide down the reader bar under the program mat for the translattion to SD card on the microcomputer via wifi
(4) Click a mouse for confirmation the executing program information
(5) Push the button on the robot for starting the program

In order to avoid the error in reading, we set sound feedback when users operate the reader bar at operation 3. Only at the operation 5, visual impairments need assistance for operation.

3. PRO-TAN

3.1 System configuration

To increase the accessibility of the tangible programming tool, we developed Pro-Tan, which has similar system configuration of P-CUBE. Pro-Tan consists of a program panel, programming cards, and a PC. Users are able to create programs controling LED light brinking and buzzer sound on Arduino ethernet. We consider that users can use Pro-Tan without instruction of the way of the system operation before programming. Fig.3 shows the concept of Pro-Tan.

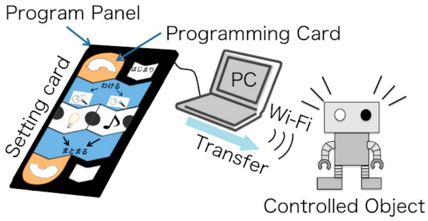

Fig. 3. Pro-Tan concept

3.2 Program panel

The program panel is made from an expanded polystyrene boards and a magnet sheet, and has ten frames which have ASI4000 RFID readers under the panel. The program panel can be attached on the wall, since the size of the program panel is 350mm×130mm, that is smaller and lighter than the program mat of P-CUBE.

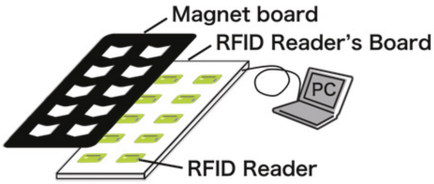

Fig. 4. Program panel

3.3 Programming card

Programming cards have four types cards as with programming blocks of P-CUBE. All programming cards are made from expanded polystyrene boards and stainless plates. Users can attach programming cards on the program panel easily using magnetic force. Since programming cards does not have tactile information like P-CUBE, the use of visually impaired indivisuals are not considered. Fig.5 shows all types of programming card on the program panel.

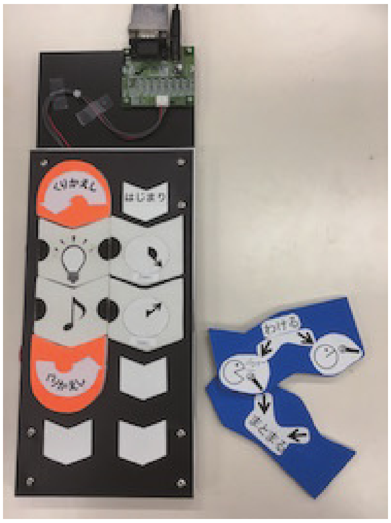

Fig. 5. Programming card and program panel

Operation card An operation card have a RFID tag on back side and instruct the movement of a LED light or a buzzer on Arduino ethernet. Users combine the concavo-convex shape of other cards when positioning cards on the panel.

Timer card Timer cards set the movement duration of the robot on a scale of one to four. Users have to set a timer card next to an operation card.

IF card IF cards correspond to the conditional branch functions depending on the information from an ultrasonic distance sensor or a microphone. The card has two RFID tags on back side and consists of begin and end cards.

LOOP card LOOP cards correspond to while loop functions. The card has a RFID tag on back side and consists of begin and end cards. A LED light or a buzzer repeats the movements indicated on cards positioned between begin and end cards.

3.4 Programming operation

Users can complete a program by positioning programming cards on the program panel adequately. Operation sequence of Pro-Tan is as follows.

(1) Select an adequate programming cards
(2) Position programming cards on the program panel
(3) Push the button on the software for traslation via wifi
(4) Push the button on the robot for starting the program

We consider that operations of Pro-Tan are easier than those of P-CUBE.

4. CHARACTERISTICS AS A GIMMICK

4.1 Effectiveness of P-CUBE

We conducted the training experiment to evaluate the effectiveness of tangible programming tool for programming beginners(9).

Experimental outline We set 2 programming tools in the experiment. One is P-CUBE as a tangible tool, and another is Arduino Sketch(14) for comparison. 7 subjects (21-14 years old) were participated in the experiment and were indicated to approach programming exercises using an assigned training programming tool. We separated subjects into 2 groups P_{tr} and C_{tr} based on the training tool. The groups P_{tr} and C_{tr} used P-CUBE and Arduino Sketch at approaching the programming exercises, respectively. 5 subjects were assigned to group P_{tr}, and others were assigned to group C_{tr}. After the programming exercises, they approached an evaluation task using both programming tools. Fig.6 shows experimental flow. We set task 1 and task 2 as programming exercises and task 3 as an evaluation task. Tool 1 is the assigned training tool and tool 2 is another tool which was used in an evaluation task.

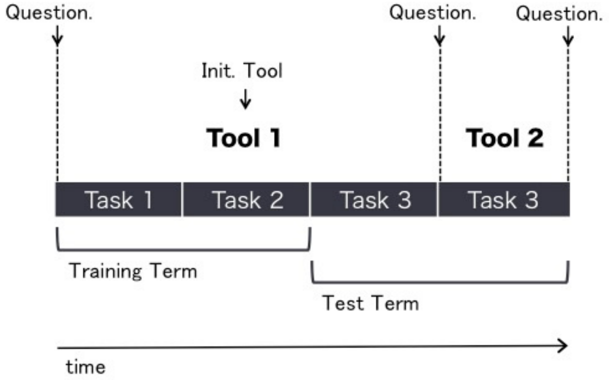

Fig. 6. Experimental flow

Tasks The experiment set 3 tasks as programming exercises.

Task1: A subject makes a sequential program which moves a mobile robot in assigned direction.
Task2: A subject makes a linetrace program which uses one IR sensor on a mobile robot. The course of task2 is showed in Fig.7 (a)
Task3: A subject makes a linetrace program which uses two IR sensors on a mobile robot. The course of task3 is showed in Fig.7 (b)

Subjects allowed to reference the documentation of each programming tools in all terms of the experiment when they lose the operation method of programming tools. We recorded the time required and the reference number of the documentation in each task. All subjects completed tasks witout assistance of staff in both terms in the experiment. Fig.8 shows experimental view.

Results The time required for the task 1 and task 2 are showed in Tab.1, and Tab.2. The average of time required for the task 1 and the task 2 of P_{tr} was shorter than those of the C_{tr}.

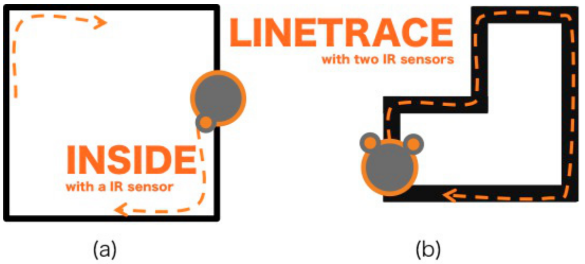

Fig. 7. Line trace course in task2 and 3

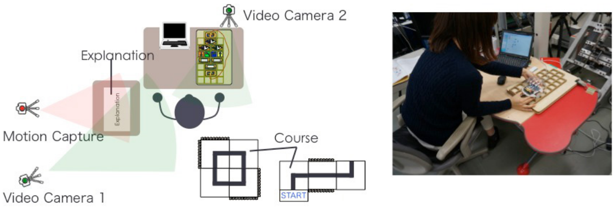

Fig. 8. Experimental view

The average of the time required in task 1 are P_{tr}:188sec, and C_{tr}:486sec, and in task 2, the average are P_{tr}:369sec, and C_{tr}:773sec. There are significant difference between P_{tr} and C_{tr} ($p < 0.05$) in both tasks.

Table 1. The time required in Task 1 & 2 - P_{tr}

Sub.	Task 1	Task 2	Total
P_1/Beg.	02:40	03:01	05:41
P_2/Beg.	02:14	02:36	04:50
P_3/Beg.	03:11	10:19	13:30
P_4/Beg.	05:24	05:45	11:09
P_5/Beg.	02:11	09:05	11:16
Ave.	03:08	06:09	09:17

Table 2. The time required in Task 1 & 2 - C_{tr}

Sub.	Task 1	Task 2	Total
C_1/Beg.	08:07	13:25	21:32
C_2/Beg.	08:05	12:20	20:25
Ave.	08:06	12:53	20:59

When the code-type tool (Arduino Sketch) was used in the task 3, the average of the completion time of P_{tr} was shorter than those of C_{tr}. There are significant difference between P_{tr} and C_{tr} ($p < 0.05$). The reference number of 6 subjects when using P-CUBE were less than the case of using the code type tool.

Table 3. The requested time and reference times of Task 3

		P-CUBE		Code Type	
Sub.	Group	Time	Ref. Number	Time	Ref. Number
P_1	P_{tr}/Beg.	06:03	0	08:31	11
P_2	P_{tr}/Beg.	07:45	0	11:35	8
P_3	P_{tr}/Beg.	13:43	9	08:36	21
P_4	P_{tr}/Beg.	05:40	0	07:20	26
P_5	P_{tr}/Beg.	06:54	5	11:23	16
C_1	C_{tr}/Beg.	09:41	4	11:40	17
C_2	C_{tr}/Beg.	16:44	15	15:17	14
P_{tr} Ave.		08:01	2.8	09:29	13.4
C_{tr} Ave.		13:13	9.5	13:26	10.5

The subjects of P_{tr} are thus considered to be able to smoothly switch to using the code-type tool and understand the fundamental program structure better than subjects of C_{tr}.

It was thus demonstrated that P-CUBE encourages beginners to understand the structure of a program.

Comments of subjects After completing each task, subjects were requested to complete a questionnaire. A number of subject comments are shown below.

- P-CUBE is easy.
- It takes a little time to understand the operation of P-CUBE.
- It is difficult to separate each sensor's IF blocks.
- I enjoyed P-CUBE programming like a game.
- It is difficult to understand the program structure when I use the code type tool.

4.2 Characterinstics of a tangible tool

We showed that using P-CUBE at an early stage of programming education allows learner to translate to code type programming software smoothly(9). In addition, P-CUBE does not require a lot of reference of the system documentation. It means that P-CUBE has the potential of becoming an easy-to-use tool which users can use at any time without usage instructions.

Since P-CUBE and Pro-Tan adopt a sytem of RFID which does not need any complicated operation like a conecting, these tools have high robustness for rough operations of younger indivisuals. This means that both tools have potentials for using in not only the classroom but any place where is not under the supervision of an instructor.

We consider that Pro-Tan has the same characteristics as P-CUBE, because it has same system configurations and operations.

Furthermore, these tools have less limitations than other tangible programming tools(1; 2; 3; 4; 5; 6), because they do not use any camera systems or wire communications. Pro-Tan uses a magnet for attaching the programming card on the program panel, program panel is able to be located in the vertical wall. Pro-Tan has high degree of freedom for placement. Fig.9 shows a program panel attached on the wall.

P-CUBE has a tactile information on the programming block for visually impairments. Therefore, P-CUBE is easy to use for collaborative learning with visual impairments. We intend to conduct an experiment to verify the effectiveness of a tangible programming tool for collaborative learning.

5. DISCUSSION AND CONCLUSION

We developed P-CUBE and Pro-Tan as a tangible programming tool for beginners in programming. We described configurations of these systems using RFID and operations in the programming. Easiness and highly robustness of these tools are indicated because of simple structures using RFID. Then, we showed characteristics as a gimmck for programming. P-CUBE and Pro-Tan have

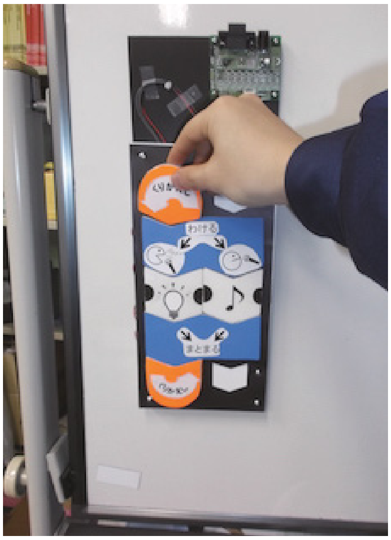

Fig. 9. Program panel on the wall

potentials for using at a place where is not under the supervision of an instructor. Especially, Pro-Tan has highly freedom for placement. This is an important element as a gimmick for programming education.

We will conduct an experiment to verify the usefulness of a tangible programming tool for collaborative learning. In addition, we intend to verify whether younger indivisuals can use Pro-Tan without preliminary instructions of operations.

REFERENCES

[1] D.Wang, Y.Zhang, S.Chen, E-Block: A Tangible Programming Tool with Graphical Blocks; *Proc. of Mathematical Problems in Engineering & MPE '13*, Vol.2013, Article ID 598547, 2013.

[2] M.S.Horn, E.T.Solovey, R.J.Crouser, R.J.K.Jacob, Comparing the Use of Tangible and Graphical Programming Languages for Informal Science Education; *Proc. of the SIGCHI conference on Human Factors in Computing Systems & CHI '09*, pp.975–984, 2009.

[3] H.Suzuki, H.Kato, Interaction-level support for collaborative learning: AlgoBlock An open programming language; *Proc. of the first international conference on Computer Support for Collaborative Learning & CSCL '95*, pp.349–355, 1995.

[4] A.C.Smith, Symbols for children's tangible programming cubes: an explorative study; *Proceedings of the 2009 Annual Conference of the Southern African Computer Lecturers' Association & SACLA '09*, pp.105–109, 2009.

[5] E.Schweikardt, M.D.Gross. roBlocks: A Robotic Construction Kit for Mathematics and Science Education; *Proc. of the International Conference on Multimodal Interaction &ICMI' 06*, pp.72–75, 2006.

[6] O.Zuckerman, S.Arida, M.Resnick. Extending Tangible Interfaces for Education: Digital Montessori-inspired Manipulatives; *Proc. of the SIGCHI conference on Human Factors in Computing Systems & CHI'05*, pp.859–868, 2005.

[7] H. Ishii and B. Ullmer, Tangible Bits: Towards Seamless Interfaces between People, Bits, and Atoms; In*Proc. of CHI'97*, pp.234-241, 1997.

[8] T.Motoyoshi, S.Kakehashi, K.Koyanagi, H.Masuta, T.Oshima, H.Kawakami, P-CUBE: Block type tool supporting programming education for visually impaired; The journal of robotics society of japan, Vol. 33. No.3, pp.172-180, 2015

[9] T.Motoyoshi, S.Kakehashi, K.Koyanagi, H.Masuta, T.Oshima, H.Kawakami, Verification of the effectiveness of P-CUBE at an early stage of program learning; Japan Society for Fuzzy Theory and Intelligent Informatics, Vol. 27. No.6, pp.909-920, 2015

[10] E. W. Dijkstra, *Structured Programming, In Software Engineering Techniques, B. Randell and J.N. Buxton, (Eds.), NATO Scientific Affairs Division*, pp.84-88, 1970.

[11] C.Bhm, G.Jacopini, *Flow diagrams, turing machines and languages with only two formation rules. In Communications of the ACM, Vol.9, No.5, ,ACM*, pp.366-371, 1966. DOI=10.1145/355592.365646 http://doi.acm.org/10.1145/355592.365646

[12] N.Matsumura, R.Fruchter, L. Leifer, Shikakeology: designing triggers for behavior change, AI & Society, Vol. 30, Issue 4, pp 419-429, 2015

[13] https://www.vstone.co.jp/products/beauto_rover

[14] https://www.arduino.cc/en/tutorial/sketch