Numeric Key Programming: Programmable Robot Kit for both Visually Impaired and Sighted Elementary School Students

Yoshihiko Kimuro\(^{(1)}\), Takafumi Ienaga\(^{1}\), and Seiji Okimoto\(^{2}\)

\(^{1}\) Fukuoka Institute of Technology, 3-30-1, Wajiro-Higashi, Higashi-ku, Fukuoka-shi, Fukuoka-ken 811-0295, Japan
kimuro@fit.ac.jp

\(^{2}\) Fukuoka Special Needs Education School for the Visually Impaired, 114, Ushijima, Chikushino-shi, Fukuoka-ken 818-0014, Japan

Abstract. In the informational society, it seems that elementary school students should learn programming and a robot kit such as LEGO is used as one of adequate programming materials. However, almost all programming tools for beginners employ graphical user interface, so visually impaired students cannot use such programming tools. To reduce the problem, we have proposed a new programming material only using a numeric keypad and a mobile toy robot. In this paper, we show the architecture of our programming environment. And we had experimental classes, which were focused on the ease of use it for both visually impaired and sighted students. As a result, visually impaired students were able to obtain the programming skill within 15 min at maximum from their first touch of the robot. On the other hands, sighted students spent only 5 min to use the robot.

Keywords: Visually impaired · Programming · Robot kit · Elementary school

1 Introduction

In the informational society, it seems that elementary school students should learn programming so that they come to know what they can do with a computer and what they cannot. However, almost all programming tools for beginners employ graphical user interface \([1]\), so visually impaired students cannot use such programming tools easily. To reduce or resolve the problem, many researchers have studied on programming languages, programming environments and programming materials for the visually impaired \([2]\).

A. Hadwen-Bennett et.al reported their work, which was based on the result of surveys on published scientific international journals in the view point of accessibility of programming environment for the visually impaired \([3]\). They classified
these papers into four categories; i.e. Text-Based Language (TBL) accessible, Block-Based Language (BBL) accessible, Physical artefacts, and Auditory and haptic feedback. However, most studies were only for visually impaired. Namely, there is no aspect that both visually impaired and sighted students can learn programming together.

J.D.Oliveria et.al. focused on using robots in programming education and surveyed relating many papers [4]. As a result of the survey, they pointed out the effectiveness of robots and 34 recommendations on robot programming for visually impaired. Unfortunately, it was more difficult to use the robot such as “LEGO” for visually impaired than sighted. Otherwise, authors introduced an original robot system “P-CUBE” proposed by Motoyoshi, which used physical blocks with RFID tags for programming. However, the special robot for visually impaired could not be used by sighted because there was no additional merit to use it instead of LEGO. Therefore, the accessibility of a robot for visually impaired should be as simple as for sighted.

2 Our Idea

First, we have to say our aim. The aim is not only to teach programming but also to teach the principle of the computer. Namely, the computer consists of a processor and a memory, as a minimal component, and a program stored in the memory is loaded into the processor and the program is executed. As a result, the computer can work according to the program. If the program includes some mistakes, the computer wrongly works according to the mistakes. If human cannot make a program, the computer cannot work. To learn these facts, students make a computer program by themselves, we think.

As an easy programming tool for beginners, visual programming languages such as “Scratch” have been employed. Such language can reduce a syntax error and a typing mistakes. However, our methodology is just the opposite of a conventional programming language for beginners.

Our aim is as follows.

1. Both visually impaired and sighted students can use a programmable robot with the same accessibility.
2. In order to teach a concept of a robot (a computer), our programmable robot is able to accept wrong program codes produced by typing mistake or syntax errors.

Figure 1 shows the overview of our mobile robot, which has a small keypad. Our robot programming system belongs to Text-Based Language (TBL). Needed keys for programming are only ten numeric keys, which are from 0 to 9. And two keys are used for “RUN” and “Reset” commands. Using these twelve keys, our mobile robot can be programmed.
3 Specification of Our Programmable Robot

3.1 Command Set

A command set of our robot has five basic commands, which are motion- and sound-related, and three flow control commands, which are “For”, “If” and “While” statements (Table 1).

![Overview of programmable robot “KOROBO SP”](image)

The basic commands consist of “opcode” and “operand”, that is a two words command. Using a one digit number, namely from “0” to “9”, working time of transition and rotation of the robot can be programmed. In the sound command, a number means a musical scale. Moreover, if “0” is set at a certain basic command as an operand, the command works as not stop command but random motion.

As flow control commands of our mobile robot, there are three statements, i.e., loop (for), conditional branch (if), and conditional loop (while). The loop procedure executes sequential statements sandwiched with “FOR n” and “NEXT” by n times. In the case of “FOR 0”, this FOR block behaves as an infinite loop. The conditional branch is described by a block structure with “IF condition” and “ENDIF”. A conditional expression consists of identifier of sensors and the state ON/OFF.

Figure 2 shows some sample programs. It is easy for everyone to understand the codes. In these figures, programs are written in English, but learners need not to remember these command names. The details will be explained in the following subsection.
Table 1. Robot programming commands

| Category | Command | Parameter | Description                     |
|----------|---------|-----------|---------------------------------|
| Basic    | FW      | d         | Move forward (0:random)          |
|          | BK      | d         | Move backward (0:random)         |
|          | LR      | d         | CCW rotation (0:random)          |
|          | RR      | d         | CW rotation (0:random)           |
|          | BEEP    | d         | Musical scale (0:random)         |
| Control  | FOR     | d         | Loop block (d:times, 0:inf)      |
|          | NEXT    | -         | End of loop                      |
|          | IF      | Sensor ID & State | Conditional branch block |
|          | ENDIF   | -         | End of IF block                  |
|          | ENDWHILE| -         | End of WHILE block (with IF)     |
| Operation| RUN/STOP| -         | Execution/temporal stop          |
|          | RESET   | -         | Clear program memory             |

![Fig. 2. Sample programs](image)

3.2 Programming Method

Our employed keypad is the most significant part of our programmable robot material. The keypad is just the same as buttons layout of a phone which is common in the world and visually impaired students can understand easily and quickly. This is because the visually impaired often use the phone than sighted students. Accessibility of the keypad for visually impaired is only a convex part on the “5” key.

The robot commands “forward”, “backward”, “turn left”, and “turn right” are located at in front, in back, at left and at right of a center key “5” respectively (Fig. 3). Namely, key layout directly means motion commands. The key “5” is used for sound command in our system.

The flow control commands are located at the rest of the above keys. For example, “FOR” and “NEXT” are set at “1” and “3” keys respectively. “IF” and “ENDIF” are set at “7” and “9” respectively. The position of keys for choosing sensors corresponds to the actual sensor position of the mobile robot. “RUN/STOP” button and “Reset” button are set at the right and left of “0”. 
3.3 Implementation

The base of our mobile robot (Fig. 1) is a toy robot (MR-9172) released by EK-JAPAN co. This robot kit has two motors, two IR (Infra-Red) sensors, two touch sensors, and a buzzer. We added Arduino UNO microcontroller board and a numeric keypad to the base robot kit.

A firmware for a command interpreter explained at the Sect. 3.1 and 3.2 is installed on the Arduino board. The program codes input from the keypad are automatically stored on EEPROM (Fig. 4). So there is no SAVE command. Now, our firmware can store 250 program steps at maximum. The depth of nesting of each FOR, IF, WHILE is 10.

4 Experiment

Using our mobile programmable robots, we had an experimental class at Fukuoka Blind Elementary School. Participants were two pupils aged 8 and four pupils aged 10. The number of totally blind was 2 and low vision was 4. The time length of the programming class was about 45 min. Each participant used one robot by themselves. As an additional material, we prepared the reflate explaining the layout of commands in braille and huge fonts.
First, participants knew a structure of the robot with their hands. Next, they learned “Reset” and “RUN” buttons. They could understand beeping when they push keys. After that, they pushed “2” and “5” key successively, then pushed “RUN”. As a result, all visually impaired were able to know the behavior of the robot caused by their own program. After participants knew the “Move forward” command, they could find the other commands “backward”, “turn left”, “turn right” experimentally.

We set a video camera to grasp how long the visually impaired students needed to obtain the programming skill (Fig. 6(a)). From the video, all students can make a sequential code and execute it on the robot within 15 min at maximum from their first touch of the robot (Fig. 5). On the other hand, sighted people also tried this programmable robot at the other place. The 127 participants were from kindergarten students to adults. As a result, almost all participants can make a sequential program within 5 min from they touched the robot for the first time. Moreover, we confirmed that visually impaired junior high school students could learn until conditional branch within two hours. Figure 6(b) shows that they made a guide dog robot which can avoid obstacles.

![Graph](image)

**Fig. 5.** The mean time of acquisition in age of subjects. Note: There is no participant aged 14.

As the results of these facts, we consider that inclusive computer education, which is for students who have or don’t have visually impaired could be achieved by using our programmable robot kit.
5 Conclusion

We proposed a new programmable robot and its programming system. Our new programmable robot can be used for both visually impaired and sighted. Further, we explained the robot command in English in this paper, but the command does not depend on English. Always the mother language of each student can be used to learn programming. This is because each command is assigned into one button from “0” to “9” so students can make command’s names by themselves.

As a future work, we will apply our 10 key programming system onto a programmable drone. In this plan, we focus on how visually impaired can know a behavior of the drone caused by a program.

References

1. Bau, D., Gray, J., Kelleher, C., Sheldon, J., Turbak, F.: Learnable programming: blocks and beyond. Commun. ACM 60, 72–80 (2017)
2. Ludi, S., Reichlmayr, T.: The use of robotics to promote computing to precollege students with visual impairments. ACM Trans. Comput. Educ. 11, 20 (2011)
3. Hadwen-Bennett, A., Sentance, S., Morrison, C.: Making programming accessible learners with visual impairments: a literature review. Int. J. Comput. Sci. Educ. Sch. 2(2), 3–13 (2018)
4. Damasio Oliveira, J., de Borba Campos, M., de Morais Amory, A., Manssour, I.H.: Teaching robot programming activities for visually impaired students: a systematic review. In: Antona, M., Stephanidis, C. (eds.) UAHCI 2017. LNCS, vol. 10279, pp. 155–167. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-58700-4_14