DEVELOPMENT OF A PROGRAMMING EXPERIENCE SYSTEM BY USING OPERATION CARDS

[1]SATJAPONG MEEKLAI, [2]KAEDE OBARA, [3]KANTAPORN SRIPRAPRUTCHAI, [4]YUKI HIRAI, [5]KEIICHI KANEKOA

[1,4,5]Graduate School of Engineering, Tokyo University of Agriculture and Technology, Tokyo, Japan
[2]Faculty of Engineering, Tokyo University of Agriculture and Technology, Tokyo, Japan
[3]Faculty of Information and Communication Technology, Mahidol University, Bangkok, Thailand
E-mail: {yhirai, k1kaneko}@cc.tuat.ac.jp

Abstract- In this research, we have designed and developed a programming experience system. The target users of the system are 1st and 2nd graders of elementary schools as well as infants who do not have programming experience. We aimed to achieve two goals, that by using our system: (1) users can become interested in programming without using keyboards and (2) they can select appropriate commands according to a situation by encouraging them in understanding the relationship between commands and their execution results. In addition, our system was implemented by using the camera function of a smartphone and augmented reality (AR) technologies. In this system, users can program by using operation cards without keyboards. We conducted a preliminary experiment to evaluate the efficacy of our system. In the experiment, with 9 children who were 1st or 2nd grade elementary students, we investigated the degree of achievement of the children who played in a game environment using the operation cards constructed in the real world. Experimental results revealed that (1) our system may be effective in terms of achievement of our research purpose, and (2) the children can be categorized into several types to make it easier for our system to support users' programming activities according to their types.

Index Terms- Programming education, Education and learning support system, Childhood education, Image recognition

I. INTRODUCTION

A. Programming Learning

Since 2000, the world has been driven by IT (Information technology), and tons of software and hardware have been developed by software engineers, hardware engineers, computer scientists, and many other types of technicians. Programming languages are gaining popularity every year in every part of the world. In 2005, Information Processing Society of Japan (IPSJ) [1] declared certain goals about information education in Japan to be achieved. One of the most important goals among them was to make people understand the mechanism of information processing through their experiences and to be equipped with the ability to utilize ICT (Information and Communication Technology) effectively. Especially, on young people, another significant goal was to encourage elementary, middle, and high school students to have experiences of appropriate 'procedural automatic processing' according to their developmental stages. Experiences of 'procedural automatic processing' include programming education [1].

In order to push IT beyond the boundary, the world needs more young people who possess programming knowledge and skills. However, to make young people have interests in programming is not an easy task because the original style of coding process is not designed to influence and get attraction from the youth. As a result, many new approaches have been proposed to solve this problem. For example, Scratch [2], the approach which is based on manipulation of objects and tiles, allows users to program their own interactive stories, games, and animations by combining provided tiles on their PCs (Personal Computers).

Scratch is designed especially for ages 8 to 16. Many primary and middle schools in many countries use Scratch to teach students programming which the students can learn important strategies for solving problems, designing projects, and communicating ideas, as well as also have fun at the same time. With respect to the above proposed approaches, because the world is now in Post-PC era, people are gradually changing from using PC to smartphones, tablets, and other technologies. The potential of using these Post-PC era technologies as another approach will deliver a new kind of programming learning experience to young people. In this way, it may be possible to boost the interests of young people in programming and also improve their analytical and problem solving skills at the same time.

B. Related Works

There are related approaches which use the technique of objects and tiles manipulation. For instance, Scratch [2], Squeak [3], OpenBlocks [4], PEN [5].

In Japan, few related studies have been conducted on primary school students. Mori et al. [6] conducted an experiment by using Scratch to offer programming classes to primary school students. Moreover, Fukaya et al. [7] also have worked on performing programming classes by creating some teaching materials using a programming environment called 'PROGURAMIN'. In contrast, these systems are designed and developed for programming by using a desktop or laptop PC. Students are likely to have some difficulties in some situations to use the above mentioned systems.

C. Purposes of This Research

Regarding the above description, we aim to give a new kind of programming experiences to students, especially 1st and 2nd grader elementary students. We believe that giving simple basic concepts of programming with the fun of problem solving challenges for young kids will help them to improve their logical thinking and popularity of programming at the same time.

We realized that there are several existing systems which are related and similar to our system. Instead of just focusing on manipulating objects and tiles, running on PC or web browsers, and following the given instructions, we designed our system to be portable and accessible anytime and anywhere. Moreover, users can freely solve our challenges without any instruction to follow in order to let them considerably develop in problem solving.

Similarly, we also would like users to think that programming is not hard, complicated, and boring. Therefore, the implemented system is a game on Android platform where millions of users include parents, teachers, and students can gain access to it. We also use the concept of augmented reality which allows users to interact with the real-world environment where real-world objects become digitally manipulable, make the game to be more attractive, easier to understand and funnier. The system can be run on either smartphones or smart glasses. As a result, users can achieve basic concepts of programming and new programming experiences as intended.

II. SYSTEM DESIGN

In this system, there are both software and hardware which need to be prepared in order to use the system.

A. Software Requirements

The system uses a SDK called Vuforia to provide augmented reality experiences to users. The SDK must be downloaded and installed in Unity. Vuforia is a vision-based augmented reality software platform that offers the widest set of features and capabilities, giving developers the freedom to extend their visions without technical limitations. With support for iOS, Android, and Unity 3D, the Vuforia platform allows you to write a single native app that can reach the most users across the widest range of smartphones and tablets. Vuforia SDK can be used to build Android and iOS applications for mobile devices and digital eyewear. Apps can be built with Android Studio (Java/C++), XCode (C++), and Unity, the cross-platform game engine.

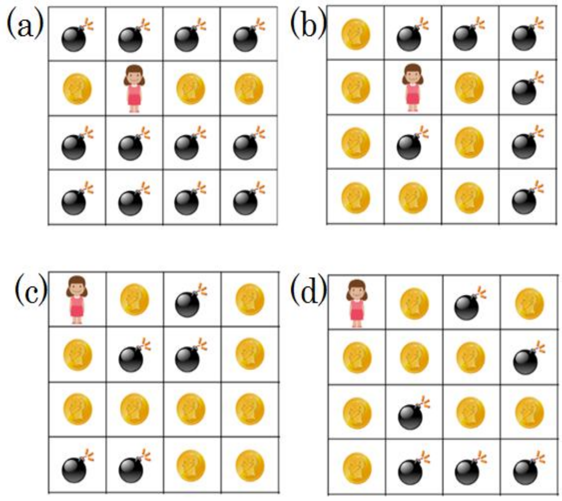

Figure 1 Game maps. ((a) Level 1, (b) Level 2, (c) Level 3, (d) Level 4)

B. Hardware Requirements

Because one of our purposes is to provide a programming learning system which does not require keyboard to input and be accessible anytime and anywhere, the system is implemented on Android platform, and designed to be a game instead. As a consequence, an Android phone with version 4.0 or later is needed. Next, we also run our system on smart glasses called Epson BT-200. However, the operation system of the smart glasses is Android 4.0. So the system can be installed and used in any kind of Android device.

C. System Design

The ideas of the system are how users can learn to program something efficiently and develop their own problem solving skills and abilities. Also, how users can keep enjoying our system is another important point that we must ensure. Thus, we considered the creation of a programming learning game by adding the concept of augmented reality to satisfy our requirements.

The concept of the game is very simple and basic. There will be a map which consists of coins, bombs and a character in each level. The maps in the game have 4 levels which are represented in Fig. 1. The map in each level is designed to be a table of size 4x4. Kids have to use provided operation cards to control the character to collect coins as much as possible within just 10 cards in each level. Kids can gradually learn types and meaning of the operation cards from basic to advance according to the game levels.

After kids either finish the level by collecting all the coins or using up all of their chances, the system will evaluate the kids' playing efficiency and effectiveness and display them before continuing to the next level. On the other hand, if the character was controlled to hit the bombs, the game will be automatically over and users need to restart the level again.

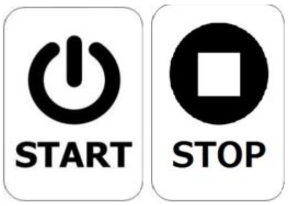

Figure 2 Game control cards (starting card (left) and stopping card (right)).

Figure 3 Character control cards.

The result of kids' playing efficiency and effectiveness is evaluated from the number of used operation cards and the number of collected coins. If users could achieve the shortest path (the minimal number of operation cards that users can use to gather all the coins), they will receive full marks.

In this game, there are 14 operation cards in total. These operation cards are separated into two groups. The first group of cards is called the game control card group, and the second group is called the character control card group. Fig. 2 illustrates the starting and stopping cards which belong to the game control card group. The starting card is used to start a game. After a user puts this card, the game interface will be displayed. The stopping card is used when the user wants to stop the game. Fig. 3 shows the character control cards. These cards are used to control the character in a game. There are 3 kinds of character control cards in this group as following:

- Fig. 3 upper row: Single move cards, the character will move one step from the current position to the direction respect to the arrow in the card.
- Fig. 3 middle row: Double move cards, the character will move two steps from the current position to the direction respect to the arrow in the card.
- Fig. 3 lower row: Jump cards, the character will jump over the object of the direction respect to the arrow in the card.

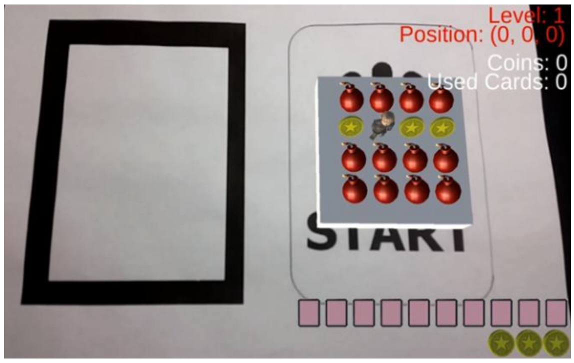

Figure 4 State immediately after game starting.

In particular, each level of the game has the following different set of character control cards that can be used:

- Level 1: Only left and right single move cards
- Level 2: All single move cards
- Level 3: All single move cards and all double move cards
- Level 4: All single move card, all double move cards, and all jump cards

The reason why we will not let users freely use all the character control cards is because our main target users of the system are 1st and 2nd grade elementary school students. Therefore, we considered the appropriate difficulty of each game level so that it will not be too complicated when kids start to play the game. Moreover, they can slowly absorb and increase their programming experiences through each level in this game.

III. SYSTEM OVERVIEW

In this section, we explain usage of our system and states of our game which consists of 4 states in total.

A. Usage of System

After users have installed our game on their smart glasses (Epson BT-200) or Android device (version 4.0 or later and has a camera function), users can start the system by opening the installed game on their Android device. If users are using the smart glasses, they should make sure that the smart glasses adjustment and position are set properly. Users should feel comfortable when they are wearing the smart glasses. Most importantly, for optimal performance of the smart glasses camera, users should position themselves about 30cm away from the operation cards.

B. Pre-running State

After the system has been opened, the camera screen will appear. Next, in order to start the game, users need to put the starting card into the camera screen. The map of level 1 will be displayed immediately after the camera detects the starting card, and then the game is started. Fig. 4 shows that the map is displayed above the starting card.

C. Game Running State

Now, the game has started. Users need to use the character control cards to move the character to collect the

Figure 5 System's response when users used a disallowed card.

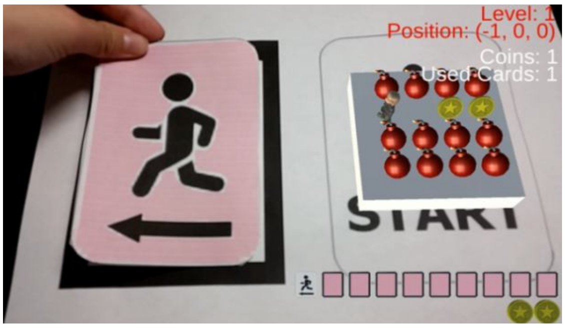

Figure 6 System's response when users got a coin.

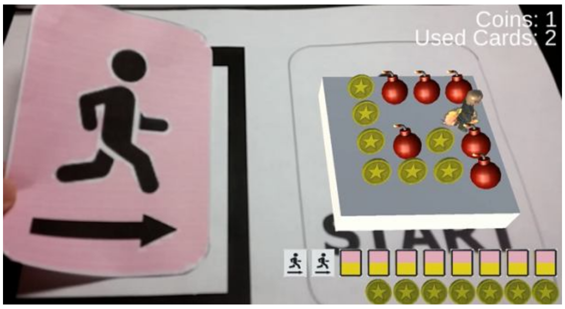

Figure 7 Game-over state.

coins as much as possible. Note that users can only use up to 10 operation cards excluding the game control cards. They are also free to use any character movement card that is allowed in a certain level. For example, in level 4, users might choose to use a double move card instead of a single move card. When users have finished each level, the evaluation of that level will be shown. After that several seconds, the next level map will be displayed automatically.

In the lower right corner of the screen in Fig. 5 and Fig. 6, there are 10 displaying little boxes and some coins. The purposes of these boxes are to illustrate the character control cards which are already in use and guiding users what character control cards are allowed in a certain level. For example, if the color inside these little boxes in Fig. 5 and Fig. 6 is pink, the system indicates that only pink cards from Fig. 3 can be used in the level.

In contrast, if users are trying to use invalid cards of a certain level, the response shown in Fig. 5 will be displayed. The system will not count the invalid cards as used cards. If valid cards are detected, the character will properly move according to the type and direction of the cards. Afterwards, the system replaces the used card to the leftmost little box in the lower right corner as shown in Fig. 6.

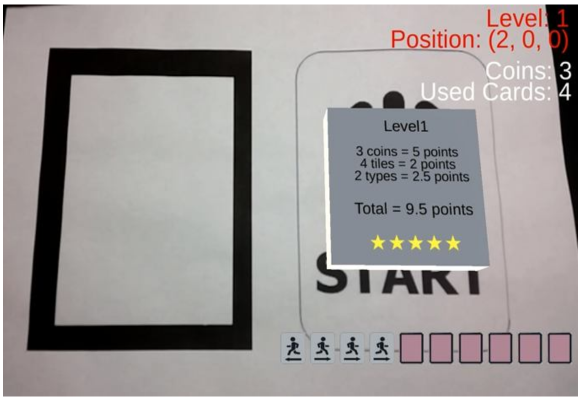

Figure 8 Displaying scores and evaluation result of a certain level.

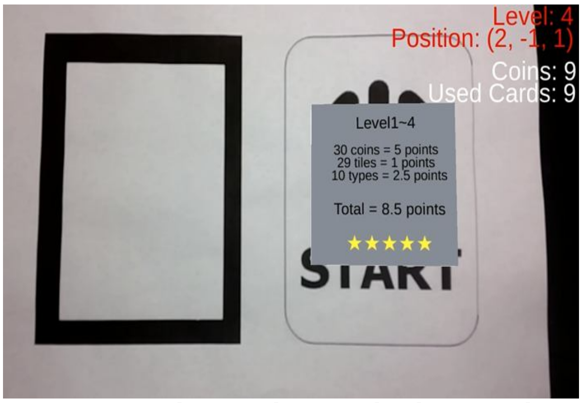

Figure 9 Displaying total scores and evaluation result.

Similarly, whenever a coin has collected, the system will also update the remaining coins of the level at the lower right corner of the screen.

D. Game-over State

There are 2 cases that the game can reach to the game-over state. The first case is when the character steps over the bomb as shown in Fig. 7. The second is when the character goes out of the map. In both cases, the map will be cleared and the game will automatically return to the pre-running state when the game is over. Users will need to start the game from the beginning again.

E. User Evaluation State

Finally, when users have finished the game in each level, the system will evaluate their playing efficiency and effectiveness by using the amount of collected coins, used cards, and types of used cards as 3 main factors. However, all of them do not have the same level of priority. The amount of collected coins plays the most significant role by 50% of our evaluation method. Next are the types of used cards and the amount of used cards which play the same priority. Our evaluation method summarizes users' scores into the form of range of 1 to 10 for the ease of understanding.

Fig. 8 shows the user evaluation after finishing each level. Fig. 9 illustrates the overall evaluation of users that will be displayed when they have finished all the levels. The purpose of overall evaluation is to give them some ideas of how well they have done in all the levels.

Table 1 Results of 6 maps provided in pre- and post-investigation.

Partici-pant	Pre-investigation		Post-investigation	
	Rate of getting coins (%)	Total # of using cards	Rate of getting coins (%)	Total # of using cards
A	52	32	42	23
B	72	40	98	52
C	33	28	78	45
D	83	50	98	55
E	75	40	76	47
F	10	8	94	54
G	83	53	88	45
H	6	5	72	40
I	54	34	78	50
Ave.	52.0	32.2	80.4	45.7

Table 2 Results of 3 maps provided in both pre- and post-investigation.

Partici-pant	Pre-investigation		Post-investigation	
	Rate of getting coins (%)	Total # of using cards	Rate of getting coins (%)	Total # of using cards
A	40	13	20	6
B	80	20	96	26
C	32	10	100	28
D	84	28	100	29
E	80	23	72	23
F	0	0	92	26
G	84	24	100	26
H	4	1	60	18
I	40	13	68	24
Ave.	49.3	14.7	78.7	22.9

IV. PRELIMINARY EXPERIMENT

We conducted a preliminary experiment to evaluate whether or not we could achieve the following two goals by using our proposed system: (1) users could become interested in programming without using keyboards and (2) they could select appropriate commands according to a situation by encouraging them in understanding the relationship between commands and their execution results. In this section, an outline and results of the experiment as well as our discussion are described.

The evaluation experiment was conducted with 9 children who were 1st or 2nd elementary school graders. In the experiment, we constructed a game environment and operation cards in the real world. This is because infants and lower-class elementary school students take much time to use the camera function of a smartphone, and they may have an illness (e.g., giddiness) by using an AR system. Note that we used blocks instead of bombs.

The experiment consisted of the following 4 sessions:

1. Explanation of games and operation cards, and practice of using the cards (10 minutes)
2. Pre-investigation (10 minutes)
3. Game (20 minutes)
4. Post-investigation (15 minutes)

Table 3 Each type's results for 6 maps provided in pre- and post-investigation.

Type	Pre-investigation		Post-investigation	
	Ave. rate of getting coins (%)	Ave. total # of using cards	Ave. rate of getting coins (%)	Ave. total # of using cards
I	55.0	32.7	96.7	53.7
II	31.0	22.3	76.0	45.0
III	70.0	41.7	68.7	38.3

Table 4 Each type's results for 3 maps provided in both pre- and post-investigation.

Type	Pre-investigation		Post-investigation	
	Ave. rate of getting coins (%)	Ave. total # of using cards	Ave. rate of getting coins (%)	Ave. total # of using cards
I	54.7	16.0	96.0	27.0
II	25.3	8.0	76.0	23.3
III	68.0	23.3	64.0	18.3

In the first session, we explained the purpose of games and how to use operation cards with some demonstration. The game purpose is that participants move a character by character control cards and get coins as much as possible. Then, we also explained the relationship between the cards and their corresponding behavior. In the second session, participants selected the character control card one by one and got coins in 6 maps provided by us. Then, we recorded participant-selected cards. Note that participants went on playing their games even if they selected invalid cards, including the cards which could signify a game over. In the third session, participants played games from level 1 to level 4 (Figure 1) with us. In these games, we notified participants anytime they selected invalid cards. Moreover, they were able to return to level 1. In the final session, participants selected cards and got coins in the same manner as the second session. Note that 3 out of 6 were the same maps provided in the second session. After that, we asked them about the difficulty of games, their impression, and so on.

V. RESULTS

A. Pre- and Post-investigation

Table 1 shows the results of pre- and post-investigation. Table 2 shows the results of 3 maps provided in both pre- and post-investigation. In tables 1 and 2, we summarized rates and total numbers in case of selecting valid cards. From the results, we can see that most of the participants got more coins and used more correct cards in the post-investigation than in the pre-investigation.

B. Participant's type

From our observation in the game session, we were able to categorize participants into the following 3 types.

- Type I: Participants who were able to get all coins without any error (Participants B, D, and F)
- Type II: Participants who were not able to get all coins in 10 turns although they did not make an error (Participants C, H, and I)
- Type III: Participants who sometimes tried to use invalid cards (Participants A, E, and G)

Tables 3 and 4 show the results for each type. From these results, Type-I and Type-II participants obtained better results in the post-investigation than in the pre-investigation; whereas, Type-III participants who did not master how to use character control cards finished with worse results in the post-investigation. Note that some of Type-II participants returned to level 1 in the game session.

C. Interview

We asked all the participants about a question: "Did you enjoy playing the games?" in the final session. All the participants said yes. Next, we asked them about the difficulty of games. As a result, we obtained the following responses:

- Easy (6 participants)
- Normal (1)
- A little difficult (1)
- Difficult (1)

Moreover, we asked them about a question: "Is there any card which you want to use besides 12 character control cards provided by us?" The following responses were obtained:

- Cards to move one step diagonally (8 participants)
- Cards to move two steps diagonally (1)
- Cards to jump diagonally (1)
- Cards to move one step diagonally and move one step vertically or horizontally (1)
- Cards to move two steps vertically and one step horizontally (like Knight in chess games) (1)
- Cards to change a block into a coin (1)

VI. DISCUSSION

A. About Our Research Purpose

With regards to the first purpose, where we tried to find out whether users could become interested in programming without using keyboards, we were able to conclude that the purpose was achieved because all the participants enjoyed playing the games and because most of participants wanted to use new cards other than our provided cards.

Next, we discuss the second purpose, which is relates to whether users could select appropriate commands according to a situation by encouraging them in understanding the relationship between commands and their execution results. As mentioned in Section 4.1, when participants selected invalid cards, we immediately notified them. That is, we encouraged them in understanding the relationship by this information. However, from the results shown in Tables 3 and 4, encouraged participants (i.e., Type-III participants) finished with worse results in the post-investigation. Therefore, such encouragement may not have any effect in understanding the relationship. As for support for Type-III participants, we intend to carry out more investigation as a future work.

In this research, we conducted the experiment with a game environment constructed in the real world. Therefore, there is the possibility that we obtain different results by conducting an experiment with our proposed AR system. In such experiment, the difficulty of letting our system recognize operation cards may occur because infants and lower-class elementary school students may not know the proper usage of the camera function of a smartphone. As a result, they may not be interested in our system.

B. About Support According to Participant's Type

As mentioned in 4.2.2, Type-II participants were not able to get all coins in 10 turns. However, they sometimes returned to level 1 because they noticed that they could not get all coins. As shown in Tables 3 or 4, Type-II participants obtained better results in the post-investigation. Therefore, retrying had some effects in understanding the relationship between commands and their execution results.

In this research, we conducted a preliminary experiment with 9 participants. If we had used a larger sample size, it would have been possible to discover new participant's types and their characteristics.

REFERENCES

[1] Information Processing Society of Japan, "The Suggestion for Japanese Education of Information and Information Processing in 2005 (in Japanese)," 2006.

[2] M. Resnick, J. Maloney, A. Monroy-hernadez, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman, and Y. Kafai, "Scratch: Programming for All," Communication of ACM, vol. 52, no. 11, pp. 60-67, 2009.

[3] A. Kay, "Squeak Etoys, Children & Learning," Writings of Viewpoints Research Institute, 2005.

[4] R. V. Roque, "OpenBlocks: An Extendable Framework for Graphical Block Programming Systems," Master thesis, Massachusetts Institute of Technology, 2007.

[5] T. Nishida, A. Harada, R. Nakamura, Y. Miyamoto, and T. Matsuura, "Implementation and Evaluation of PEN: The

Programming Environment for Novices (in Japanese)," Journal of Information Processing Society of Japan, vol. 48, no. 8, pp. 2736-2747, 2007.

[6] H. Mori, M. Sugisawa, H. Zhakg, and T. Maesako, "Practical Study on Scratch Programming Lessons for Elementary School Students: Rethinking Programming Education at Elementary School (in Japanese)," Journal of Japan Society for Educational Technology, vol. 34, no. 4, pp. 386-394, 2011.

[7] K. Fukaya and A. Miyachi, "A Study on the Utilization of 'PROGURAMIN' for Programming Lessons for Elementary School Students (in Japanese)," Journal of Japan Society for Educational Technology, vol. 36, no. suppl., pp. 9-12, 2012.

★★★