International Journal of Advanced Smart Convergence Vol.7 No.4 128-137 (2018)
http://dx.doi.org/10.7236/IJASC.2018.7.4.128

IJASC 18-4-16

On the Effect of a Pilot Coding Education Support System for Complex Problem Solving Tasks

Inseong Jeon* and Ki-Sang Song †

**Gwangju Educational Research Information Service*
† Dept. of Computer Education, Korea National University of Education, Korea
**wf212@naver.com, † kssong@knue.ac.kr*

Abstract

In the programming education, there is a great need of a teaching support system that can support the learner in the programming process regardless of the computer language due to instructor's difficulty of checking the progress of learners in real-time. Its importance is especially important in lower grade coding classes such as in K-12 education because they are not used to coding and so simple problems can be regarded as complex problems. For this, a pilot coding education support system based on Levenshtein distance algorithm which shows learners' progress to given solution in real-time was developed in order to help learners to solve complex problems easily, and the learners' motivation and self-efficacy was measured for estimating the usefulness of developed system targeting elementary school students. When the learners use the developed system, it was found that a statistically significant difference appears in the sub-factors of learning motivation compared with traditional class teaching environments. Among the sub-factors of self-efficacy, the efficacy dimension showed statistically significant difference too.

Key words: Coding Education, Complex Problem Solving, Learning Motivation, Self-efficacy

1. Introduction

A report on the advent of the fourth industrial revolution, presented at the World Economic Forum (WEF) in 2016 [1] shows that the revolution in the technology sector can fundamentally change the way we work with our way of life as well. To prepare the 4th industrial revolution, which is expected to bring about such drastic changes, the executives of 370 companies in 15 countries such as United States, China, Japan, Germany, Britain and France [2] considered the complex problem solving skill as the most needed talent for the future.

Complex problems that companies encounter are problems caused by various changes in market conditions. In order to be able to deal with these problems, it requires the ability to solve problems that are not clearly defined or unstructured in complex and realistic environments [3]. To develop these complex problem-solving abilities, it is important to understand the problem based on solid critical thinking practices, to look at other factors influenced by the problem and other factors affected by the problem, and to find the best solution

Manuscript Received: Oct. 17, 2018 / Revised: Oct. 25, 2018 / Accepted: Oct. 28, 2018
Corresponding Author: kssong@knue.ac.kr
Tel: +82-10-6402-8644, Fax: +82-43-231-4790
Dept. of Computer Education, Korea National University of Education, Korea
*Some part of this paper has been presented at the International Conference on Computational Thinking Education 2018, 14-16 June 2018 at Hong Kong.

among alternative solutions ability.

Although the importance of raising students with complex problem solving abilities in school education is emphasized in every dimension, K-12 education has limitations of dealing with ill-structured problems. For example, teachers are requested to follow designed curriculums and those subjects mainly deal with well-defined problems in general. As digital technologies penetrate into every aspect of life, coding education that mainly focusing on computational thinking (CT) has prevailed in the world. The coding education in K-12 education may handle ill-structured problems, and hence force it may provide the opportunity of training students to acquire complex problem abilities. However, teaching coding to K-12 students needs well-designed support systems due to the difficulty of checking the progress of learners in real-time.

Therefore, this paper describes the effectiveness of a pilot coding education support systems for helping elementary school teachers who try to introduce complex problem solving ability through coding education. The final goal of this study is to develop a learning analytics system that can help learners to guide complex problem solving process by observing learner's coding process. Up to now we have used the developed pilot system, and assess the effectiveness of the developed system in terms of learner's motivation and self-efficacy while solving complex problems in schools by applying 10 classes at elementary school coding education.

2. Background

2.1 The Need for Intelligent Programming Support Systems

As AI technology evolves, there is a wide variety of applications in education. Efforts to provide intelligent and adaptive feedback by applying data mining techniques have been continuing since the 1980s, mainly to utilize computers as a tutor. If such an intelligent support system is developed, it can be helpful for beginners and learners who are not familiar with programming such as in coding education.

For beginners, especially in lower grades students, it is difficult to express commands in English, which can lead to a lot of syntax errors in text-based programming environments. In this way, learners not only experience frustration but also teachers have many difficulties to grasp the directions of learners. To overcome this problem, block-based programming language such as Scratch and Entry have been developed as educational programming languages for K-12 education and non-CS major students.

These languages seem to overcome the disadvantages of text-based programming environments much in that they allow learners to create programs by dragging blocks of commands, such as assembling blocks. Nevertheless, there remains a problem that it is not easy for the instructor to understand the progress of programming until the final result of the task is presented. For this reason, teachers who are teaching coding education in elementary schools in Korea experience heavy burden of handing many students' questions, and supporting students at the same time in their class. Therefore, the necessity of teachers to provide appropriate feedback by grasping the progress of learners' coding process in real-time is still greatly required in the block-based programming environment [4].

2.2 Related Studies

In programming education, showing programming results utilizing computers to students are mainly implemented in the form of 'online judge'. Online Judge is an online system designed to test programs at programming competitions. It is also used for practice exercises of learning algorithms, where the system compiles and runs the code, and tests it with pre-written data. The submitted code may be subject to restrictions on time, memory, security, and so on. The system receives the output data, compares it with the standard output, and gives the result. This system is utilized in lessons such as algorithm education, etc. However, the online judge does not have the ability to provide real-time feedback data to the teachers, so there is a limit to providing learning support to the instructor. Some of the well known foreign online judge sites include Hackerrank, and Codeforces etc.

Adopting the online judge idea, Kim designed and implemented a web-based programming task evaluation system that allowed the teacher to automatically evaluate the performance of the program and to easily check the style and plagiarism of the program with appropriate feedback [5]. Song designed and developed an automatic scoring based programming education system that could perform learner-centered self-directed learning by performing programming education more efficiently [6]. Chang & Kim [7] developed a client-server based system by enforcing teaching-learning functions of the existing online judge style system and found its significant effect on programming learning. Jeong [8] developed a system that can be used for programming learning and evaluation of science high school students, unlike Online Judge system which is used for the evaluation in the competitions.

Unlike these online judge type systems, Oh, et. a.l. [9] implemented a system to support teachers in a text-based programming environment. They developed a program that allows students who are programming in text-based languages environment to perform detailed student learning activities such as modifying the code. The systems capture detailed student learning activities, such as modifying the code so that these data allow the system to assess student progress and use it to design and improve course materials and structures.

However, there have been few studies related to the development of a system that supports block-programming languages for elementary and secondary school students and teachers up to now. Price and Barnes [4] argue that even though iSnap [10] may give hints to students when they stuck during programming, it is necessary to develop intelligent tutoring system (ITS) for block-programming environments. Therefore, in this study, we use a pilot version of coding education support system that performs learning analysis function with a block-based programming language and report the test results of the system on learning motivation and self-efficacy of learners.

3. Proposed pilot coding education support system

In this pilot system [11], a learner can program using Entry, a block-based programming language, through a web server, and click the save button to analyze his/her source code in real-time comparing with teacher's solution.

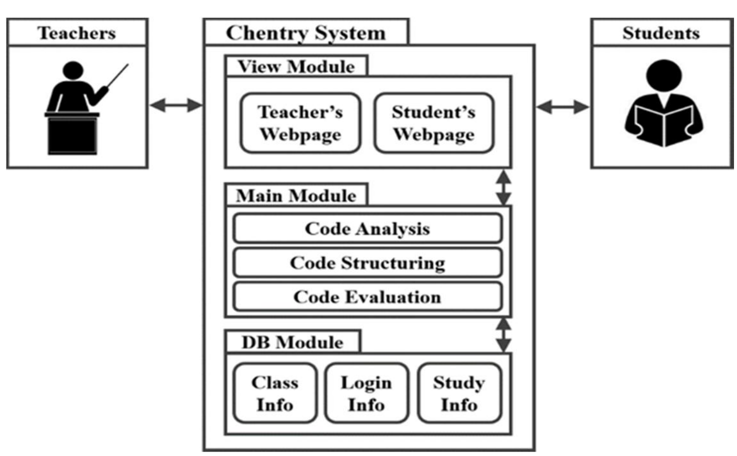

Figure 1. Modules of developed pilot system

Unlike Price and Barnes' argue, to understand a learner's programming progress, it is not necessary to have all the features of a traditional ITS components such as learner model, hint or feedback module. The system and the teaching and learning model that we want to develop focus on observing the programming process, and finding the learners who need guidance a timely. Therefore, we devised the learner model with learner profile often adopted in the conventional ITS, the correct answers to the given tasks presented by the instructor, and functions of periodically storing and analyzing the learner's coding process.

To this end, the main module has been developed to perform the following procedures. The teacher opens a class by entering the class name and the URL address of the answer code. The learner accesses to the system with his/her account, enters the URL address of the source code, and created programs in Entry environment. The system checks whether the learner clicks the save button at a pre-determined time interval, and saves accumulated achievements using the Levenshtein distance algorithm when the save button is clicked.

Levenshtein distance is a kind of edit distance technique that calculates the minimum number of edits such as deletions, insertions, or substitutions required when a string is converted into another string [12].

$$\mathrm{lev}(s_1, s_2) = \frac{\mathrm{dist}(s_1, s_2)}{\max(|s_1|, |s_2|)} \quad (1)$$

In this study, the progress of students' learning was calculated through individual block group agreement and whole block group agreement [11].

$$\text{Individual block group agreement}(\%) = 100 \times \left(1 - \frac{Levenstein\ block\ distance}{MAX(No.of\ Teacher\ block, No.of\ student\ block)}\right) \quad (2)$$

$$\text{Whole block group agreement}(\%) = 100 \times \left(1 - \frac{\Sigma Levenstein\ block\ distance}{\Sigma MAX(No.of\ Teacher\ block, No.of\ student\ block)}\right) \quad (3)$$

In addition, if the block type is the same but only the variable or the block parameter is wrong, it is considered to be corrected 0.5 times rather than 1. The teacher can check which block each learner has used, how the block of the learner has been changed for a certain time, and what the degree of final achievement is. The teacher can also provide corrective feedback to the student with low achievement level. The algorithm of the overall pilot system through the above agents is shown in Fig. 2.

4. Assessment of learners' motivations and self-efficacy with pilot system

4.1 Research Method

For verifying the educational effects of the pilot system developed in this study, an experiment was administered on 40 students of grade 5 students in S elementary school in Kwangju, Korea. The experimental group and the control group received a total of 10 units of programming education using block-programming language environments. In the case of the experimental group, the teacher identified the progress of the learners with the developed system, intervened appropriately, and let the learners compare their progress with the teacher's answer though computer display. For the control group, on the other hand, the traditional teaching method was used where the teacher roamed around the classroom to observe and give advice to the learners

■ Coding Subjects for the Test

The curriculum of 10 classes applied to the both experimental and control group was consists of learning computer science concepts such as sequence, repetition, selection, simple variable, and list. Those topics are set up by the Ministry of Education of Korea taught in elementary school software education subjects from 2019. The summary of the curriculum is shown in Table 1.

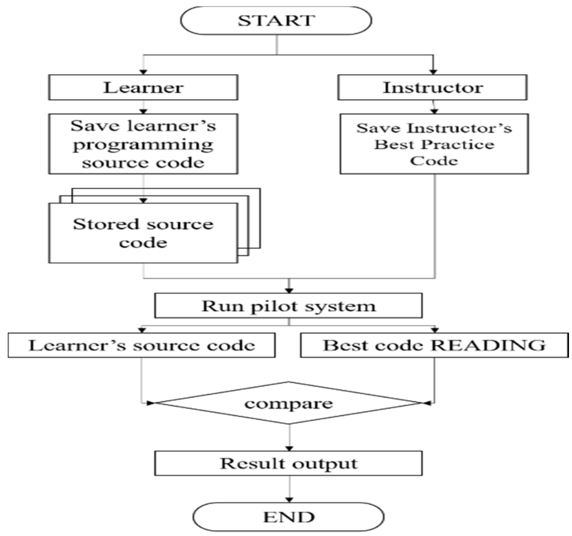

Figure 2. The algorithm of the pilot version of coding education support system

Unlike traditional text-based programming courses, the block-based programming environment provides an easy game creation environment. For example, many game development environments such as Scratch developed by MIT or Entry provide abstractions of programming logic components, graphics-oriented, user-friendly adoption, and interfaces that allow developers to easily configure their game logic [13]. Thus, learners create coding as if they are making games. This is a very important point of applying block-based programming environment for K-12 coding education. This plays a very important role in training complex problem solving abilities. Since the complex problem solving procedure is implicitly applied in the game design process, and therefore game design learning (GDL) [14] is a good way of training students to acquire complex problem solving abilities.

Table 1. Tasks given to each class

Time	Title	Elements
1	Basic programming language manipulation Use movement, shape block	concatenation
2	Create simple block application program	concatenation, repetition
3	Make 'bears meet bees' Use repetition with variables	concatenation, repetition, variables
4	Make 'Bee shot bear' Use selection Structures	concatenation, repetition, selection
5	Make 'Shark Avoid' Game	concatenation, repetition, selection
6	Develop games: Use Replication Blocks	concatenation, repetition, selection, variables
7	Complete the game: Use lists and variables	concatenation, repetition, list, variables
8	Make a gift lottery program Use random number	concatenation, repetition
9	Make 'Producer Speaker' Use Random Numbers, Arithmetic Expressions	concatenation, repetition, list, variables
10	Draw a polygon Use pen-block, basic functions	concatenation, repetition, function

Figure 3. Two example tasks for creating code

Figure 3 shows two snapshots of tasks assigned to students. The scenario for the left figure [When bear meets bee] is as follows. When the bear of the scene meets bee, follow the mouse pointer controlled by the user, and needs to shout “There is a thief!,” until hitting the left or right wall and if bear hits the wall needs to shout “Ah, it’s bee!.” All given tasks consist of unique scenario as games, and learners were asked to create programs with Entry to perform given story with animation, conversations between actors according to pre-defined rules. Creating code that has game characteristics may help students to improve complex problem solving abilities [14].

■ Teaching and Learning Methods for Both Group

In the experimental group, learners were allowed to check their learning achievement through the pilot system constantly during the 10 units of classes. The teacher checked the achievements of learners during the class and provided corrective feedback to the student who kept having low achievement level quite a long time by analyzing the causes of low achievement. In the control group, the teacher gave a lecture just like in the conventional teaching method and gave feedback directly to the learners while roaming around the classroom.

■ Motivation Test and Self-Efficacy Test Inventory

In order to measure the learning motivation of the students, the motivation test tool for youth developed by Lee and Jung [15] was modified for the elementary school students. The learning motivation test tool was composed of 26 questions in total; specifically 5 items of Amotivation, 5 items of External Regulation, 5 items of Introjected Regulation, 5 items of Identified Regulation and 6 items of Integrated Regulation. Amotivation is the status that a desire to learn is not generated regardless of external stimuli, External Regulation is behavior control by external factors, Introjected Regulation is to act through the influence of past experiences such as reward and punishment, Identified Regulation means that the integrated control as the external factor is changed into an internal factor, and the Integrated Regulation means the motivation to create and achieve something on its own.

In coding class, it is important learner has not only a motivation to learn but also the belief that the learner can do well for the task. The belief that they can do well in coding education stems from the ability to handle the programming language rooted in the personal experience of programming activities and the level and strength of their experiences. Therefore, we used the self-efficacy test contains three sub-factors; ‘Language’ to measure their understanding of computer programming, ‘Self-efficacy factor’ to show their direct and indirect experience of handing computer programming to solve complex problems, and ‘Self-efficacy dimension’ to measure the intensity of programming activities.

For measuring the self-efficacy of students in the programming language, the self-efficacy test tool in computer programing language education environment developed by Kim [16] was modified. The self-efficacy test tool is composed of 30 questions in total; specifically 10 questions about language, 10 questions about Efficacy Factor and 10 questions about Efficacy Dimension.

4.2 Result

In order to verify the homogeneity of the experimental group and the control group, a pre-test was administered to measure learning motivation and self-efficacy of two groups. As Table 2 shows, there was no significant difference, which confirms the homogeneity of the groups.

Table 2. Homogeneity Test of Groups for Measuring the Learning Motivation and Self-Efficacy

Area	Group	N	M	SD	t	P
Learning Motivation Total	Experimental	20	73.90	7.873	-.421	.676
	Control	19	75.16	10.658		
L1	Experimental	20	22.50	4.274	.432	.668
	Control	19	21.79	5.903		
L2	Experimental	20	10.25	3.226	-.934	.357
	Control	19	11.32	3.888		
L3	Experimental	20	12.20	3.122	-.959	.344
	Control	19	13.26	3.784		
L4	Experimental	20	10.95	3.900	-.401	.691
	Control	19	11.53	5.037		
L5	Experimental	20	18.00	3.356	.580	.565
	Control	19	17.26	4.520		
Self-efficacy Total	Experimental	20	93.20	16.979	.165	.870
	Control	19	92.11	23.949		
S1	Experimental	20	29.75	7.813	.024	.981
	Control	19	29.68	9.310		
S2	Experimental	20	30.90	5.330	-.814	.421
	Control	19	32.84	9.167		
S3	Experimental	20	32.55	6.362	1.343	.187
	Control	19	29.58	7.434		

$* p < .05$

L1=Integrated Regulation, L2=Amotivation, L3=Introjected Regulation, L4=External Regulation, L5=Identified Regulation S1=Language, S2=Efficacy Factor, S3=Efficacy Dimension

The results of the post-test on learning motivation and self-efficacy are shown in Table 3. There were significant differences in Integrated Regulation, External Regulation and Introjected Regulation among the sub-factors of learning motivation, but there was no significant difference in Amotivation and identification control. There was a significant difference in Efficacy Dimension in the sub-elements of self-efficacy, but there was no difference in Language and Efficacy Factor.

Table 3. Post-test of Groups for Measuring the Learning Motivation and Self-Efficacy

Area	Group	N	M	SD	t	P
L1	Experimental	20	22.30	4.567	.2074	.044*
	Control	19	19.33	4.902		
L2	Experimental	20	10.74	3.374	-1.623	.112
	Control	19	12.43	3.529		
L3	Experimental	20	12.57	4.660	-.997	.324
	Control	19	13.86	3.851		
L4	Experimental	20	10.78	4.680	-1.863	.069
	Control	19	13.38	4.555		
L5	Experimental	20	18.65	3.688	2.308	.026*
	Control	19	15.76	4.603		
S1	Experimental	20	33.35	7.352	.6511	.519
	Control	19	31.86	7.844		
S2	Experimental	20	37.04	8.054	.650	.106
	Control	19	33.24	7.162		
S3	Experimental	20	34.70	8.578	1.994	.050*
	Control	19	30.05	6.659		

$* p < .05$

L1=Integrated Regulation, L2=Amotivation, L3=Introjected Regulation, L4=External Regulation, L5=Identified Regulation S1=Language, S2=Efficacy Factor, S3=Efficacy Dimension

In order to analyze the difference between the self-efficacy before and after the application of the system in the experimental group, the mean and the standard deviation were calculated by dividing the test period. Table 4 shows the paired t-test results. As presented in the table, the sum of self-efficacy after the system application is significantly higher than before the system application. In the analysis of sub-factors, there was a statistically significant difference in Efficacy Dimension and there was no difference in Language and Efficacy Factor.

Table 4. Paired Samples t-Test of Group to Measure for Self-Efficacy

Area	Group	Paired Differences			t	P
		N	M	SD		
Total	Pre Post	20	14.10	28.10	2.244	.037*
S1	Pre Post	20	4.20	11.91	1.578	.131
S2	Pre Post	20	6.60	8.18	3.606	.002*
S3	Pre Post	20	3.30	11.32	1.303	.208

S1=Language, S2=Efficacy Factor, S3=Efficacy Dimension $* p < .05$

5. Conclusion

Even though Price and Barnes [4] proposed a block-based programming environment with intelligent tutoring system for individual learners, but teachers need a system to monitor learners progress in real-time for providing feedback with easier and more effective way in school classrooms. For this purpose, we developed an entry-based teaching assistant system and the pilot system proposed in this study can be an example of a realistic method of computer-aided coding education support system. As it was appeared in self-efficacy sub-factor test, students were able to get successful experience of learning in the coding process of given complex problems through the developed pilot system. Learners' positive responses to the experience of dealing with complex problems or data through the computer programming process will naturally enhance the use of computers in dealing with complex problems in the future. Also, it is shown that the learning method through the proposed pilot system can prevent students from losing their motivation or experiencing negative experiences in classrooms that solve complex problems.

Acknowledgment

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government (Ministry of Education) (2017S1A5A2A01026058).

References

[1] World Economic Forum White Paper Digital Transformation of Industries: Digital Enterprise. World Economic Forum, January, 2016.

[2] The Future of Jobs - Employments, Skills and Workforce Strategy for the Fourth Industrial Revolution. World Economic Forum, January, 2016.

[3] Quesada J., Kintsch, W., and Gomez. E., Complex problem solving: a field in search of a definition?, Theoretical Issues in Ergonomics Science Vol. 6 , Iss. 1, pp.1-48. 2005.

[4] Price, T. W, and Barnes, T. “Position paper: Block-based programming should offer intelligent support for learners,” 2017 IEEE Blocks and Beyond Workshop (B&B). 2017.

[5] Kim, M. H. Design and Implementation of an Automatic Grading System for Programming Assignments. Journal of Internet Computing and Services, 8(6), pp.75-85. 2007.

[6] Song, J. H. An Automated Assessment based Programming Education System for Self-Directed Learning. Doctoral dissertation, Soongsil University. 2011.

[7] Chang, W. Y. & Kim, S. S. Development and application of algorithm judging system: analysis of effects on programming learning. Journal of Korean Association of Computer Education, 22(2), pp.15-24. 2014.

[8] Jeong, J. K. Design and Construct of Programming Assessment System based on "Online Judge" for a Science High School student. M.S. thesis, Korea National University of Education. 2010.

[9] Kim, S., Kim, J. W., Park, J.K., and Oh, A., Elice: An online CS Education Platform to Understand How Students Learn Programming, Proceedings of the Third (2016) ACM Conference on Learning @Scale, pp. 225-228. 2016.

[10] Price, T. W., Dong, Y., and Lipovac, D., “iSnap: Towards Intelligent Tutoring in Novice Programming Environments,” in Proc. of ACM SIGCSE, 2017.

[11] Kim, J,-H., Choi, J.-H., Shadikhodjaev, U., Nasridinov, A., and Song, K.S. “Chentry: Automated Evaluation of Students’ Learning Progress for Entry Education Software,” Advances in Intelligent Systems and Computing Series, Vol. 770. Big Data Applications and Services 2017, The 4th International Conference on Big Data Applications and Services, Springer, pp.51-60, 2018.

[12] Levenshtein, Vladimir I. "Binary codes capable of correcting deletions, insertions, and reversals". Soviet Physics Doklady, 10(8), pp.707–710. 1966.

[13] Du, E., Gameblox Flexidor: Adding Flexibility to Blocks Based Programming Environments, M.E. Thesis, MIT. 2015.

[14] Akcaoglu, M., Gutierrez, A., Hodges, C. B., and Sonnleitner, P., Game Design as a Complex Problem Solving Process, Handbook of Research on Serious Games for Educational Applications (Advances in Game-Based Learning), 2016.

[15] Lee, M. H., & Jung, T. Y. Development and Validation of the Learning Motivation Scale. Studies on Korean Youth, 18(3), pp.295-321. 2007.

[16] Kim, K. S. Measuring and Applying the Self-efficacy in Computer Programming Education. Journal of The Korean Association of Information Education, 18(1), pp.111-120. 2014.