S. L. Wong et al. (Eds.) (2010). Proceedings of the 18th International Conference on Computers in Education. Putrajaya, Malaysia: Asia-Pacific Society for Computers in Education

The Effects of Scaffolding-Based Courseware for The Scratch Programming Learning on Student Problem Solving Skill

Dongsoo NAM, Yungsik KIM, Taewook LEE
Department of Computer Education, The National University of Education, Korea
*namdongsoo@hanmail.net

Abstract: Programming learning is of use for the improvement of complicated problem-solving skills that modern society calls for. Yet it puts heavy strain on learners, and the careful selection of a good program tool and the improvement of learning methods are required to relieve learners of the burden. Scratch is an educational programming language that has an affluent multimedia programming setting and is easy and interesting to learn. So that is appropriate for programming beginners, and the use of a scaffolding-based courseware has a superb effect on the development of problem-solving skills in association with programming that is difficult to do. The purpose of this study was to develop a scaffolding- based courseware geared toward facilitating scratch programming learning and to examine the effects of it on problem-solving skills. As a result of utilizing a scaffolding-based courseware developed in this study, that turned out to have a significant effect on boosting the problem-solving skills of the learners who learned by using the courseware.

Keywords: Scaffolding, Scratch, Courseware, Problem Solving, Programming

Introduction

What matters in knowledge-based society is converting information into internalized knowledge, applying the knowledge to problem situations and solving the given problems. Problem-solving skills could be said to refer to solving new problems and learning new knowledge by applying existing knowledge. Indeed, problem-solving skills are one of primary mechanisms that affect the way of thinking, work and even life styles. A wide variety of studies are under way in various academic communities such as psychology, computer science and education to enhance problem-solving skills[3][4]. In cognitive science and intelligence theories, the skills are discussed from a theoretical and general perspective, and industries and the education sector put stress on more practical, down-to-earth problem-solving skills. Engineering researchers and educators view contextuality or contextual complexity as one of the primary conditions for the enhancement of problem-solving skills and provide opportunities for learners to go through situations similar to reality to bolster their problem-solving skills[7].

A computer programming process is a process of modeling or automating an abstract thinking process through computing equipment in pursuit of problem solving, and the experience of the computer programming process could lead to more reinforced abstract thinking[2][1]. That is, the programming process requires automation skills to automate abstract skills or abstract concepts to choose and organize the best abstract concepts for the purpose of problem solving[5]. Such a cognitive ability is different from problem-solving or reasoning skills stressed in other academic fields in that the former could provide the best solution by producing strategic knowledge applicable to general problem-solving situations and by creating and automating a problem-solving process through computing equipment[2]. Accordingly, the development of authentic and complex problem-solving

skills needs programming education, and research efforts should be put into selecting efficient teaching-learning tools, what to teach and the best teaching-learning methods to ensure successful programming education.
Scratch is one of teaching-learning tools that can back up programming education successfully and has a lot of benefits. That is an educational programming language jointly developed by the U.S. MIT Media Lab and UCLA researchers, and that is a graphic building-block programming language that piles up blocks whose commands are all different according to color and form on objects called sprites and programs the sprites. Learners and teachers take a dim view of existing programming education that is very time-consuming, and that educational programming language is easier to understand and learn irrespective of age and can make intuitive programming possible. Furthermore, that gives support to various kinds of multimedia and different languages. Thus, the programming language is very splendid and has a lot of benefits.
Scaffolding refers to providing temporary help for learners to learn successfully in teaching-learning process, which was introduced by Wood, et. al., who tried to apply Vygotsky's theory as one of the major elements of individualized teaching. Scaffolding denotes what more competent peers, teachers or parents give an appropriate introduction or help to learners and consequently serve as a scaffold to step up their cognitive development. In other words, that is a backup system that serves to boost the task performance or cognitive process of learners by offering visual clues, modeling or feedback without altering the given task itself and lets them determine for themselves how much help they need to perform the task on their own based on the degree of task performance.

This study intended to find out the impact of a scaffolding-based courseware on scratch programming learning by investigating whether there would be any gaps in problem-solving skills between one group that utilized the scaffolding-based courseware and the other that didn't.

1. Background

Understanding of Scratch and The Meaning of Scaffolding

Scratch is an educational language that was developed by the Lifelong Kindergarten Research Team of MIT Media Lab. That is object-oriented and based on visual graphic under the influence of Logo, SmallTalk, HyperCard, StarLogo, AgentSheets, Etoys and Tweak.
Scaffolding was introduced by Wood, Brunner and Ross[6] who tried to identify the major components of effective individualized teaching by applying Vygotsky's zone of proximal development theory. They defined it as help that adults or experts provide for children or beginners to solve the given problems, perform the given tasks and achieve their goal on their own..

2. Method

Hypothesis

The following hypothesis was formulated:
A scaffolding-based courseware designed to facilitate scratch programming learning might have an impact on problem-solving skills.

Subject

The subjects in this study were 60 students who were in their sixth grade in an elementary school located in Gangwon Province, Korea. An experimental group and a control group were made up of 30 students respectively.

Instrumentation

The scaffolding-based courseware used in this study was designed to be implemented in eight sessions. And we teach 4 weeks(2 hours per week).

The Structure of the courseware

Table 1. The Structure of the Courseware

Period	Theme
1	Deal with motion blocks
2	Deal with observational blocks
3	Deal with variable blocks
4	Deal with operation blocks
5	Deal with control blocks
6	Deal with blocks of form
7	Deal with sound blocks
8	Deal with pen blocks

Two Learning Modes of the Courseware

There are two different modes in which students can study with this courseware. One is a demonstration/repetition mode, and the other mode is doing alone. Appropriate scaffolding is provided while students study with the courseware.

Demonstration / Repetition

In this mode, the computer gives a demonstration on how to do, and then learners do the same. An error message is given whenever they make any programming mistake, and they are helped to do it again.

The Characteristics of Scaffolding

In this mode, learners perform the given task for themselves. If they make a programming error, a cue is provided(scaffolding) to help them do it again.

Evaluation Instrumentation

We use the PISA 2003 problem-solving items that were modified to be appropriate at school children's level.

Research Design and Procedure

The experimental design and procedure of the study are as below:

Table 2. The experimental design and procedure of the study

Group	Pretest	Experiment	Posttest
The Experimental Group	O1	X1	O2
The Control Group	O3	X2	O4

A pretest was conducted to evaluate the problem-solving skills of the two groups to see whether they were equivalent or not. And then the experimental group took lessons by applying the scaffolding-based courseware, and the control group took typical lessons. The experiment was implemented during a four-week period of time, and they were given tasks immediately after the experiment to find out the effects of the experiment.

3. Result

Pretest Result

A independent-samples t-test was carried out before the experiment, and there were no significant intergroup gaps at the .05 level of significance($p>.05$). So the two groups were considered to be equivalent in terms of problem-solving skills.

Table 3. Pretest Result

Classification	N	M	SD	DF	t	p
The Experimental Group	30	9.546	3.50	65	-.667	-.507
The Control Group	30	10.088	3.16			

Posttest Result

A posttest was conducted after the experiment, and the collective average of the experimental group was larger than that of the control group. As for the intergroup gaps in mean scores, the value of t was 2.375, and the probability of significance was .021, which showed that there was a significant difference between the two in problem-solving skills in the .05 level of significance. So the scaffolding-based courseware applied to the experimental group exerted a great influence on the improvement of problem-solving skills.

Table 4. Posttest Result

Classification	N	M	SD	DF	t	p
The Experimental Group	30	13.3636	3.55	65	2.375	.021
The Control Group	30	11.5	2.84			

The Result of Pretest and Posttest in Each Group

A paired-samples t-test was carried out to find out the pretest and posttest scores of each group. As seen on the table, the pretest scores of the experimental group were significantly different from the posttest scores of that group at the .05 level of significance, and the probability of significance was .000. It indicated that the students who studied with the scaffolding-based courseware made progress in problem-solving skills.

When the pretest and posttest scores of the control group were compared, there was a significant gap at the .05 level of significance, and the probability of significance was .001. So the students who studied in the typical way showed improvement in problem-solving skills as well.

Table 5. The Result of Pretest and Posttest in Each Group

Group		Pretest-Posttest				
		M	SD	DF	t	p
The Experimental Group	Pretest	9.5455	3.55	32	4.264	.000
	Posttest	13.3636	3.5			
The Control Group	Pretest	10.0882	3.16	33	3.783	.001
	Posttest	11.5	2.84			

Conclusion

The objects of programming education that aims at improving problem-solving skills are not only to acquire the use of the programming tool itself but to gain actual experiences in problem solving through the programming process. The programming process that designs a problem-solving process and configures it as a computer program is a complicated problem-solving process itself, in which learners are able to integrate a variety of knowledge and knowhow and acquire strategic knowledge to flexibly cope with problem situations. Yet the problem is that programming is difficult and time-consuming to learn. Therefore it's needed to show how to approach programming education easily, and one of the best ways is the use of scratch and the supply of scaffolding.

The findings of the study were as follows:

First, programming learning turned out to be effective. Both of the experimental and control groups showed a significant improvement in problem-solving skills, and it could be interpreted that programming learning has a positive impact on problem-solving skills irrespective of teaching methods.

Second, the scaffolding-based software turned out to be effective. Both of the experimental and control groups used the scratch when they engaged in programming learning, but the experimental group that was provided with scaffolding made a better significant progress in that regard.

References

[1] Eunkyung, Lee. Youngjun, Lee. (2008). The Effects of Programming Ability and Immersion Level for Middle School Student on Programming Education using The Scratch, The Middle School Education Research. 56(2). pp. 359-382.

[2] Eunkyung, Lee. (2009). Robot Programming Teaching and Learning Model to Enhance Computational Thinking Ability. Doctoral Thesis. Korea National University of Education.

[3] Jitgarun, K. (2002). The Development of Computer-Based Simulation Laboratory for e-Learning : A case Study of Non-Electrical Engineering Background at KMUTT in Thaillad. World Conference on E-Learning in Corp., Govt., Health., & Higher Ed. 2002(1), pp. 1669-1672

[4] Najjar, L. J. (1996). Multimediainformation and Learning. Journal of Educational Multimedia and Hypermedia, g, pp. 129-150

[5] Wing, J. M. (2006). Computational Thinking. Communications of the ACM, 49(3), 33-35.

[6] Wood, D., Bruner, J. S. & Ross, G. (1976), The role of tutoring in problem solving. Journal of Child Psychiatry and Psychology, 17(2), 89-100.

[7] Youngmin, Lee. (2005). The Effects of Simulation Delivery Instruction on Students's Problem Solvng Performance and Motivation. KACE. 8(5), pp. 43-50.