International Journal for Educational Media and Technology
2013, Vol.7, No. 1, pp. 15-25

Pre-service Teachers' Conceptions and Reflections of Computer Programming using Scratch: Technological and Pedagogical Perspectives

Hyungshin Choi
Chuncheon National University of Education, Korea

This paper presents a case of a computer programming course with a new approach for Korean pre-service teachers. Computer programming is generally considered hard to master and unproductive due to dealing with syntactical errors. As an alternative approach, Scratch was incorporated into a programming course for pre-service teachers. Scratch is an educational programming language developed by the MIT Media Lab to help beginners approach programming through drag-and-drop coding blocks and remixing various media such as images, sounds, texts or narrations. This study aims to discover how Scratch changes Korean pre-service teachers' perspectives toward computer programming and also what their artifacts reveal. The data for analysis includes surveys and reflections from 96 pre-service teachers who took the programming course in the spring semester at a national university of education in Korea. In addition, the digital learning content created by each group was collected and analyzed. The research results present pre-service teachers' conceptions and reflections of computer programming and their intentions to use it in their future teaching. From the collected artifacts, the types of digital content and the expected learning effects of teaching content were examined. This paper discusses the implications of the research findings for teacher education programs as well as pre-service teachers' adaptive expertise for 21st century learning environments.

Keywords: Scratch, EPL (Educational Programming Language), Pre-service Teachers, Higher-Order Thinking Skills, Computational Thinking

Introduction

Computer programming can facilitate the development of higher-order thinking skills such as logical thinking, creative thinking, problem solving, and systematic thinking (Fesakis & Serafeim, 2009). In addition, the process of computer programming supports students' development of computational thinking (Papert, 1993; Wing, 2006). Wing argues that computational thinking is fundamental for everyone in the 21st century and it should be added as the 4th R followed by reading, writing and arithmetic. Computational thinking involves solving problems, using heuristic reasoning, and using abstraction and decomposition (Wing, 2006). Computational thinking is also defined as "critical thinking and problem solving with an

added integral computational component" (Kranovet al., 2010, p. 144).

Computer programming is considered to be a part of digital literacy because it enables users to become dynamic producers of digital content. For future teachers, computer programming familiarization is not only for educational use but is also necessary to create interactive digital learning content.

Despite the benefits and importance of computer programming, it is considered difficult to master. This is because novice programmers have to deal with syntactical errors that they encounter as they type the commands within complex logical structures. The failure or drop-out rates in introductory computer programming courses are estimated to be 15 to 30% (Fesakis & Serafeim, 2009).

Among various approaches to reduce the difficulties of beginner programmers (Guzdial, 2004), an educational programming language (EPL) such as Scratch has been proposed. Scratch is an educational programming language developed by the Lifelong Kindergarten research group at the MIT Media Lab (MIT Media Lab, 2012). Scratch provides visual programming blocks that are composed by dragging and dropping to create programs. There are eight different types of blocks: motion, looks, sound, pen, control, sensing, operators, and variables. Through Scratch programming, novice programmers have opportunities to understand concepts such as logical structures (sequential, selection, repetition, conditional), variables and lists, event-driven processing, debugging, etc. In addition, the users also have a chance to create media rich projects such as interactive stories, simulations, and games.

The focus of previous studies has been on the affordance of Scratch to support students dealing with problems of logic rather than programming components (Malan & Leitner, 2007). In addition, other research has concentrated on the educational potential of young students' engagement with Scratch with its simplified interface and mechanisms (Maloney et al., 2008). This paper focuses on the experiences of pre-service teachers with Scratch and their perceptions and reflections on computer programming while they produce learning content using Scratch. The rest of the paper presents the theoretical background in regard to the relationship between programming and learning skills, and then describes the research methods followed by the results and discussion of the main findings and future directions.

This paper reports the current study with particular interest in the following research questions:

1. How Scratch experiences change the perspectives of Korean pre-service teachers toward computer programming in a technological aspect.
2. How Scratch experiences change their perspectives toward computer programming in a pedagogical aspect.
3. What Korean pre-service teachers' computer programming artifacts manifest.

Theoretical Background

Although computer programs are run by computers, programming is closer to thinking skills than computer skills. Learning how to program provides many benefits including acquiring computer knowledge, problem-solving skills, and thinking skills (Barr & Stephenson, 2011; Tasneem, 2012). Researchers at the National Science Foundation suggest that offering a programming course should not be limited to computer majors (NRC, 2010). This is because every career requires students to think and solve problems, and programming can teach students both.

Programming with an EPL such as Scratch is claimed to support the development of 21st century learning skills described by the Partnership for the 21st century (http://www.21stcenturyskills.org). The nine types of learning skills are divided into three key areas: information & communication skills, thinking and problem-solving skills, and interpersonal & self-directional skills (Rush, Resnick, & Maloney, 2012). First, through Scratch programming, students gain experience with multiple forms of media such as text, images, narrations, and songs. Not only do students obtain media literacy skills, but they also learn how to choose, manipulate, and integrate various media to communicate effectively. Second, Scratch engages students in systemic thinking by providing opportunities to try the concepts of sensing and feedback. In addition, Scratch requires students breaking the problem into steps and experimenting with their ideas. Third, Scratch encourages students to work collaboratively exchanging code modules and objects with peers. When students share their projects, they receive feedback from others and learn how to adapt to their ideas.

Programming with rich media promotes learning by doing and the development of pre-service teachers' adaptive ability. In the fast-changing world, pre-service teachers should be equipped to adjust in response to the complexity of the environment of the 21st century (Resnick, 2007). In order for pre-service teachers to implement adaptive expertise, the teacher education program should focus more on cultivating innovative ideas and applying them iteratively (Hammerness, Darling-Hammond, & Bransford, 2005).

Design and Methodology

Data Collection and Participants

This study is based on data collected from pre-service teachers in Korea, using a survey instrument that examines technological and pedagogical aspects of Scratch programming experiences. In June 2012, the survey was administered to third year student teachers who had taken the Computer Practice course. A convenience sampling method was used to recruit participants. In total, 96 student teachers participated in the survey. As shown in Table 1, the sample includes 23 male (24%) and 73 female (76%) participants. Among the 96 participants, 78 (81.3%) student teachers reported that they had no prior programming experiences. In addition, the qualitative data includes reflections from 94 pre-service teachers. 49 team projects were also collected.

Table 1. *Age, gender, prior programming experiences of the participants (n= 96)*

Category		N(%)
Gender	Male	23 (24%)
	Female	73 (76%)
Age	20~22	54 (56.2%)
	23-26	24 (25%)
	27~30	18 (19.6%)
Prior Programming Experience	No	78 (81.3%)
	Yes	13 (13.5%)
	No response	5 (5.2%)

Survey Instrument, Reflections and Artifacts

The survey instrument with 15 items on a 5-point Likert scale (1 = strongly disagree, 2 = disagree, 3 = neutral, 4 = agree, 5 = strongly agree) was developed to examine the perceptions of pre-service teachers about Scratch programming. The instrument includes two factors: (1) a technological aspect (9 items) and (2) a pedagogical aspect (6 items). The first four items for Scratch interface in the technological aspect were used from the previous research conducted by Fesakis and Serafeim (2009). The rest of the items were created by the author in terms of the technological and pedagogical aspects of Scratch programming. The internal reliability of each factor is listed in Table 2. The results indicate acceptable levels of internal reliability for all factors (above .70).

The reflections were free format documents that consist of 2~3 paragraphs. Each artifact included a Scratch project and a presentation slide that explained the context, subject, topic, purpose, and the expected effects of the digital content. Each team was able to choose any subject, topic, and grade within the elementary curriculum to develop the content.

Table 2. *Surveys factors and internal reliability (Cronbach's alpha)*

Factors		**No. of items**	**Cronbach's alpha**
Technological Aspect	Scratch Interface & Digital Content Integration	9	.76
Pedagogical Aspect	Creating Teaching Content	6	.76

Data Analysis

SPSS 12.0 was used to provide descriptive statistics for the survey items. For a qualitative analysis, the data (students' reflection papers) was examined to explore two categories: technological and pedagogical aspects. The author labeled incidents in the data and constantly compared them with one another to decide which of them belonged together (Strauss & Corbin, 1990). NVivo 9.2 was incorporated to handle the coding process and count the incidents in each category accurately.

Results

The Technological Aspect

Perceptions on the Technological Aspect

As shown in Table 3, participants' perceptions for the Scratch interface and the technological effects of Scratch were positive (M=3.99, M=4.00). Pre-service teachers perceived that Scratch is user friendly (M=4.25) and uses understandable language (M=4.11). In addition, they reported that Scratch made them think computer programming is not as hard as they thought (M=4.26).

Table 3. *Mean and standard deviation for the technological effects of Scratch (n=96)*

No.	Opinions about Scratch	Mean	SD
1	Scratch is user friendly.	4.25	.71
2	User interface uses a simple and understandable language.	4.11	.71
3	Scratch has aesthetic design.	3.80	.90
4	Scratch features are satisfying.	3.79	.79
	Average	3.99	.51
No.	Perceptions about the Technological effects of Scratch programming	Mean	SD
1	Scratch made me to think that computer programming is not as hard as I thought.	4.26	.68
2	Scratch made me feel more confident in integrating various digital media. (e.g., images, sounds, texts, etc.)	4.03	.76
3	Scratch made me feel that I can express more creatively with digital media.	4.11	.68
4	Scratch made me feel more comfortable with sharing and creating digital content with peers.	3.70	.94
5	Scratch made me feel more comfortable in selecting and utilizing useful digital media.	3.90	.83
	Average	4.00	.54

Reflections on the Technological Aspect

The reflections of the participants revealed that Scratch changed their attitudes toward programming. Their negative feelings such as fear, anxiety, perplexity, and difficultness were changed to feelings of fun, easiness, accomplishment and joy of discovery. In addition, pre-service teachers' expectations changed from simplicity to value and usefulness. Table 4 shows some sample reflections in each code and the numbers of incidents.

Table 4. *Code types of reflections on the technological aspect*

No.	Code	Pre-service Teachers' Reflections	# of Incident
1	Anxiety to Fun	"Although I had taken several programming courses, programming was a word seems to be quite far and difficult to me. Scratch is the one that breaks my prejudice. The realization process of what I thought just by drag-and-drop was interesting and fun. "	12
2	Difficult to Easy	"I thought programming is just difficult combinations of English and commands. But learning by blocks and images I realized that it is not as difficult and now I can design programs for my teaching goals."	13
3	Perplex to Accomplishment	"I was perplexed at the beginning. But going through several weeks of hands-on programming was fun and I was filled with pride and a sense of accomplishment."	11
4	Fear to Confidence	"I had fear toward machine and computers. All I could do with computers was browsing and creating documents. So the burden I felt as I attend this class was huge. But as I learn and program with Scratch every class I gradually gain confidence with computers."	18
5	Simplistic to Useful	"First I learn Scratch I had some doubt about how much I would use this program. But after I learned the details of Scratch programming I realized that it is useful."	21

The Pedagogical Aspect

Perceptions on the Pedagogical Aspect

As shown in Table 5, pre-service teachers' perceptions about the pedagogical effects of Scratch programming is overall positive (M=4.00). Among six items, participants perceived that through Scratch their ability to create digital learning content has improved (M=4.18). In addition, Scratch made pre-service teachers think more positively toward using ICT in education (M=4.23) However, they were not as confident teaching the fundamental programming concepts using Scratch (M=3.55).

Table 5. *Mean and standard deviation for the pedagogical effects of Scratch (n=96)*

No.	Perceptions about the pedagogical effects of Scratch programming	Mean	SD
1	Through Scratch programming, my ability to create digital learning content has improved.	4.18	.63
2	Through Scratch programming, my ability to create digital learning content that can increase students' motivation has improved.	4.23	.64
3	Through Scratch programming, my ability to create digital learning content that can increase teaching effects has improved.	4.11	.63
4	Scratch made me think positive toward using ICT in education.	4.23	.76
5	When teaching content is not proper, I would rather create the content myself using Scratch.	3.85	.91
6	I think I could teach elementary students fundamental programming concepts using Scratch.	3.55	.83
	Average	4.00	.50

Reflections on the Pedagogical Aspect

The analysis of the reflections on the pedagogical aspects revealed three different types of themes. First, the reflections of the participants showed the effects of Scratch programming in regards to innovation. The pre-service teachers realized that Scratch can help them create more interactive and context based digital learning content that reflects teachers' intentions and philosophy. Second, the reflections also revealed that pre-service teachers believe Scratch programming can help them develop thinking skills such as logical thinking, creative thinking, critical thinking, and systemic thinking. Third, the reflections represented that they saw the value and usefulness of teaching content created by Scratch programming and showed their intentions for future use (See Table 6).

Table 6. *Code types of reflections on the pedagogical aspect*

No.	Code	Pre-service Teachers' Reflections	# of Incident
1	Innovation	"It is important that teachers provide rich resources to students. Teachers should persistently think about more interesting and understandable teaching methods. Scratch is something that can help teachers put this thought into practice. "	25
2	Thinking skills	"While I was designing the final projects, I considered how I can utilize specific functions and imagined what they would produce. As I was going through this process and resolving problems, I naturally practiced creative and logical thinking. "	7
3	Intentions to use for future teaching	"Scratch has a very easy interface that a non expert someone like me can implement programs. Not only I use it for this semester, I expect to use it to create class materials when I actually teach in classes in the future. "	72

Artifacts

Summary of the artifacts

In total, 49 artifacts were collected and analyzed in terms of the target grade, subject, the type of content, and the expected learning effects. Pre-service teachers selected science (37%) the most followed by Math (19%), Language (18%), and Social Science (16%). In addition, 62% of the pre-service teachers selected 5thor 6th grade content followed by 4th (15%) and 3rd (14%) (See Figure 1).

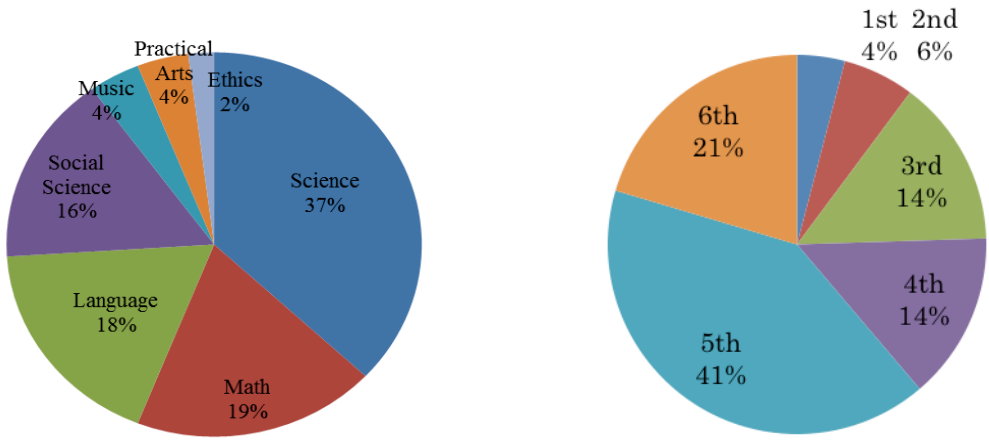

Figure 1.*The Distribution of artifacts in subjects (left) and grades (right)*

As shown in Figure 2, six different types of artifacts were manifested in 49 artifacts. The most popular type were quizzes (22, 45%) followed by presentations (9, 18%), simulations (9, 18%) and games (6, 12%). Some artifacts combined more than one type but each one was categorized by the major approach taken. The expected learning effects include evoking motivation, formative evaluations, summarization, group collaboration, process presentation, etc.

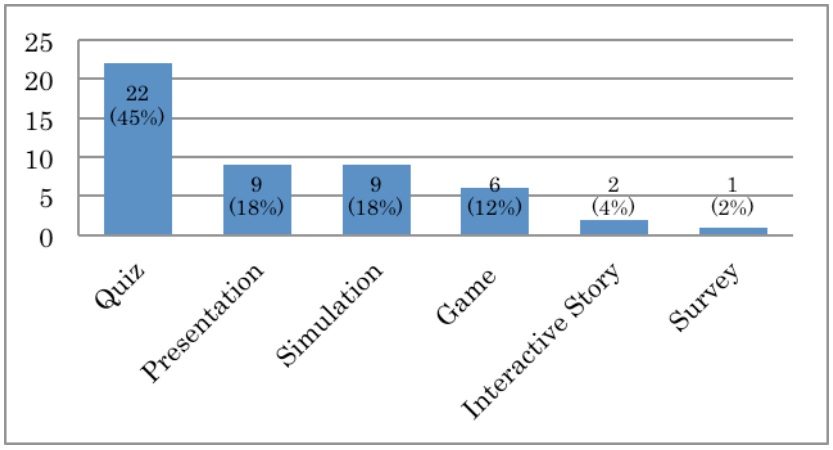

Figure 2.*The types of digital content created with Scratch*

Discussion and Conclusion

The research results of the study validate the educational decision to use Scratch in a computer practice course for future teachers. Pre-service teachers acknowledged that Scratch has a positive effect both on technological and pedagogical aspects. The reflections of pre-service teachers revealed a sense of 'innovation' that involves moving beyond existing routines and rethinking practices to change what one is currently doing (Hammerness, Darling-Hammond, & Bransford, 2005). For teachers to be adaptive experts, they need to attempt innovative strategies that seem less efficient initially and to perceive the experiences as a valuable process of learning (Kim, Choi, Han, & So, 2012).

This study, however, revealed that although pre-service teachers think positively toward using ICT in education through Scratch programming, they did not perceive that they are confident enough to teach elementary students the fundamental programming concepts. This result might be due to the fact that Scratch encourages students to learn programming concepts more implicitly than other programming languages such as C or Java. As Guzdial (2004) once pointed out, the purpose or expectations of teaching computer programming need to be reconsidered. In a Scratch environment, learning to program by manipulating various media and connecting blocks to control these media can yield fluent digital content producers. However, the students may require further trainings to become more knowledgeable to be able to teach programming concepts.

In conclusion, I argue that Scratch has positive effects, not just in a technological sense but also in a pedagogical aspect. It should be noted that this study has some limitations. First, by using a convenience sampling procedure, the selection of particular pre-service teachers could have caused an uncontrolled bias. Second, this study analyzed the participants' artifacts by the content type and the learning effects at a surface level. This work could be continued in the future to investigate evidence of the development of thinking skills such as computational thinking and logical reasoning.

References

Barr, V., & Stephenson, C., (2011). Bringing Computational thinking to K-12: what is Involved and what is the role of the computer science education community? *ACM Inroads archive*, *2*(1), 48-54.

Fesakis, G., & Serafeim, K. (2009). Influence of the familiarization with "Scratch" on future teachers' opinions and attitudes about programming and ICT in education. *ACM Inroads archive*, *41*(3), 258-262.

Guzdial, M. (2004). Programming environments for novices. In S. Fincher & M. Petre (Eds.), *Computer science education research* (pp. 127-154). London & New York: Routledge Falmer.

Hammerness, K., Darling-Hammond, L., &Bransford. (2005). How teachers learn and develop. In L. Darling-Hamoond & J. Bransford (Eds.), *Preparing teachers for a changing world* (pp. 358-389). San Francisco: Jossey-Bass.

Kim, H., Choi, H., Han, J., & So, H. J. (2012). Enhancing teachers' ICT capacity for the 21st century learning environment: Three cases of teacher education in Korea. *Australasian Journal of Educational Technology*, *28*(6), 965-982.

Kranov, A., Bryant, R., Orr G., Wallace, S., & Zhang, M. (2010). Developing a community definition and teaching modules for computational thinking: Accomplishments and Challenges. *Proceedings of the 2010 ACM conference on Information technology* (pp 143-148).

Malan, D., & Leitner, H. (2007). Scratch for Budding Computer Scientists. ACM Special Interest Group on Computer Science Education. Covington, Kentucky: ACM. http://cs.harvard.edu/malan/publications/fp079-malan.pdf

Maloney, J., Peppler, K., Kafai, Y. B., Resnick, M., & Rusk, N. (2008, March). Programming by choice: Urban youth learning programming with Scratch. Paper presented at the SIGCSE 2008 Conference, Portland, OR.

MIT Media Lab. (2012). Scratch. Retrieved on 4 July 2012 from http://scratch.mit.edu.

National Research Council (2010). Report of a workshop on the scope and nature of computational thinking. Washington D. C.: The national academies press.

Papert, S. (1993). *Mindstorms: Children, computers, and powerful ideas*. Cambridge, MA: Perseus Publishing.

Resnick, M. (2007). Sowing the seeds for a more creative society. *Learning and Leading with Technology, 35*(4), 18-22.

Rush, N., Resnick, M., & Maloney, J. (2012). Learning with Scratch 21st century learning skills. Retreived on July 2, 2012 from

http://llk.media.mit.edu/projects/scratch/papers/Scratch-21stCenturySkills.pdf
Strauss, A., & Corbin, J. (1990). *Basics of qualitative research: Grounded theory procedures and techniques*. Thousand Oaks, CA: Sage.
Tasneem, S. (2012).Critical thinking in an introductory programming course. *Journal of computing sciences in colleges, 27*(6), 81-83.
Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35.