Wang, X.-M., Hwang, G.-J., Liang, Z.-Y., & Wang, H.-Y. (2017). Enhancing Students' Computer Programming Performances, Critical Thinking Awareness and Attitudes towards Programming: An Online Peer-Assessment Attempt. *Educational Technology & Society, 20* (4), 58–68.

Enhancing Students' Computer Programming Performances, Critical Thinking Awareness and Attitudes towards Programming: An Online Peer-Assessment Attempt

Xiao-Ming Wang[1], Gwo-Jen Hwang[2*], Zi-Yun Liang[2] and Hsiu-Ying Wang[3]
[1]School of Teacher Education, Zhejiang Normal University, Jinhua, China // [2]Graduate Institute of Digital Learning and Education, National Taiwan University of Science and Technology, Taiwan, R.O.C. // [3]Department of Education, National University of Tainan, Tainan, Taiwan, R.O.C. // zsdwxm@gmail.com // gjhwang.academic@gmail.com // flora91410@gmail.com // cute@tkgsh.tn.edu.tw
[*]Corresponding author

(Submitted May 21, 2016; Revised September 21, 2016; Accepted December 11, 2016)

ABSTRACT

It has become an important and challenging issue to foster students' concepts and skills of computer programming. Scholars believe that programming training could promote students' higher order thinking performance; however, many school teachers have reported the difficulty of teaching programming courses. Although several previous studies have attempted to develop friendly user interfaces to ease students' loads, teaching programming courses remains a big challenge for most school teachers. In this study, an online peer assessment-based system was developed to cope with this problem. The students could use the peer-assessment function to provide comments to peers, and review the feedback and scores from peers during the learning activity. A quasi experiment was conducted on four classes of 166 ninth graders of a junior high school located in southern Taiwan to examine the impacts of the developed system. Two classes of students were assigned to the experimental group, learning with an online peer assessment-based teaching strategy, while the other two classes were the control group, learning with the conventional teaching strategy. The experimental results showed that the students in the experimental group had better programming knowledge and skills as well as more positive learning attitudes and critical thinking awareness than those in the control group, revealing the benefits of the proposed approach.

Keywords

Computer programming, Peer assessment, Critical thinking, Learning attitude, Scratch

Introduction

How to effectively conduct programming education to help students develop the concepts and skills of programming has become an important and challenging issue (Krpan, Mladenović, & Rosić, 2015; Yang et al., 2015; Brito & de Sá-Soares, 2014). Many researchers believe that engaging in programming tasks not only enables students to gain knowledge of programming, but also has great potential in promoting their higher order thinking performance (Keppens & Hay, 2008; Wang, Huang, & Hwang, 2016; Williams et al., 2002). However, past experience shows that many students perceive learning computer programming as a difficult and boring task (Kátai, 2015). Therefore, researchers have attempted to develop programming environments with graphical interfaces to ease students' load of developing computer programs. Scratch is such a programming environment that has been widely adopted by school teachers. According to the report of Kobsiripat (2015), who used Scratch as the programming language to let students produce their own works, it was found that the students' performance was increased due to the easy manipulation learning environment in Scratch.

Despite the advancements in programming tools and environments, teaching programming languages remains a big challenge for most school teachers (Barr & Guzdial, 2015; Sáez-López, Román-González, & Vázquez-Cano, 2016). Researchers have suggested that, to effectively promote students' learning performance in programming courses, it is necessary and important to include proper teaching strategies (Chang, Wu, Weng, & Sung, 2012). As computer programming is relevant to not only programming knowledge (i.e., knowing how a computer program works and the syntax of a programming language), but also programming skills (i.e., knowing how to analyze a task and develop a program for dealing with the task), it could be more effective to engage students in the practice of viewing and emulating with guidance from the teacher. Project-based learning with peer assessment is such an approach that engages students in playing the roles of project developer and reviewer based on the rubrics provided by the teacher. Such a learning activity provides great opportunities to create bigger learning effects for students than the conventional teaching and practice approach (Gielen, Peeters, Dochy, Onghena, & Struyven, 2010; Hsia, Huang, & Hwang, 2016; Liu & Carless, 2006). The value that peer assessment brings to education is that in the process of peer assessment, students have opportunities to learn the rubrics provided by the teacher in depth, and to learn from peers' work since they play the role of reviewer. This implies that they are situated in the context of practicing critical thinking about peers' work and providing concrete suggestions and feedback to their peers (Hovardas, Tsivitanidou, & Zacharia, 2014). In addition, while evaluating others' work, they make comparisons of their own work with that of others; hence, they are likely to ponder whether their work meets the teacher's standards. Such a learning process provides a strong chance of developing students' critical thinking awareness (Boud, Cohen, & Sampson, 1999; Liu & Carless, 2006; Topping, 1998; Xiao & Lucking, 2008).

On the other hand, several studies have also reported that peer assessment might put pressure on students due to the interpersonal relationships involved, which might lead to negative impacts (Falchikov, 2004). Vanderhoven, Raes, Schellens, and Montrieux (2012) suggested that online peer assessment could be a solution to cope with this problem. Compared to traditional face-to-face peer assessment, using computer technologies that enable students to provide feedback to peers anonymously could significantly reduce their pressure and avoid the negative impacts. However, little research has attempted to employ the online peer assessment approach for programming projects. Moreover, to the best of our knowledge, no empirical study has been conducted to show the effects of online peer assessment on students' programming knowledge or skills. Therefore, in this study, an online peer assessment system was developed and an experiment was conducted to examine students' computer programming performance using the system and the Scratch programming environment. The research questions of this study are as follows:

- Can an online peer assessment-based approach increase students' learning achievement (i.e., programming knowledge and skills) in Scratch programming?
- Can an online peer assessment-based approach increase students' learning attitude toward Scratch programming?
- Can an online peer assessment-based approach increase students' awareness of critical thinking?

Literature review

Scratch programming

Scratch is a programming language that was developed by the Lifelong Kindergarten research group of the Massachusetts Institute of Technology in 2007. Scratch was designed to help young people learn creative thinking, logistical inferencing, and collaborative learning, which are considered to be basic skills needed in the 21st century. Scratch provides users with a visual interface which allows them to join blocks together to easily create interactive stories, games, and animations on their own. It uses blocks as the objects to command. Such an innovative programming environment means beginners can avoid making syntax and logical errors (Ouahbi, Kaddari, Darhmaoui, Elachqar, & Lahmine, 2015). This method allows students to create multimedia works easily and develop their creativity and thinking ability during the learning process (Resnick et al., 2009; Wang, Huang, & Hwang, 2016). Many scholars believe that Scratch is an innovative learning method for students to be creative in this digital era (Kim & Song, 2012; Moreno, 2012; Peppler & Kafai, 2007).

In recent years, an increasing number of tools using Scratch as a programming language have been adopted in junior high and elementary schools. In Ouahbi, Kaddari, Darhmaoui, Elachqar, and Lahmine's (2015) experiment in one senior high school, it was found that using Scratch to let students program can bring out their potential for developing games; 65% of them expressed that they were willing to continue using Scratch to create, while only 10.3% of the students were willing to use the traditional Pascal programming language to program. The conclusion reached was that using Scratch to let students learn programming languages in a basic programming design course can effectively increase their learning motivation; students do not need to worry about the programming syntax, and can easily develop their games and animations. Meanwhile, many researchers have pointed out that using Scratch to teach programming can let students learn the syntax for judging and looping, and for creating their own personal works (Fesakis & Serafeim, 2009; Peppler & Kafai, 2005). Kobsiripat (2015) adopted Scratch to conduct a programming design learning activity for 60 elementary school students; it was found that through this activity, the students' creativity and higher order thinking benefited. Wang, Huang, and Hwang (2016) conducted a project-based learning activity for students with advanced and ordinary performance using the Scratch programming language; the results showed that the advanced students performed better in terms of their learning achievement, problem-solving ability, learning attitude, and learning motivation than the ordinary students.

In this study, Scratch was adopted as the programming tool owing to the following reasons. First, it provides a graphical programming interface which enables novice learners and young students to realize the basic programming logic without being confused by programming language syntax. Moreover, it has been widely adopted for a number of years now in Taiwan's high schools, including the selected school, as the learning tool in programming courses.

Peer assessment

Peer assessment, as proposed by Topping (1998), is carefully designed guidance for giving feedback to peers, and can be seen as a teaching strategy to help students review the advantages and weaknesses of their peers' work so that they can make amendments to the goals not reached while developing their metacognitive ability, critical thinking awareness, test performance, and professional skills (Joordens, Pare, & Pruesse, 2009; Topping, 2009). Peer assessment can be applied as a formative or summative evaluation by providing students with rubrics from teachers to evaluate their peers' work and to give suggestions (Liu & Carless, 2006; Topping, 1998). During the peer assessment process, students need to view works from the teacher's point of view and consider whether their peers' performance meets the teacher's requirements. On the other hand, while evaluating peers' work, students are also given a chance to reflect on and think about how to improve their work; such a process can benefit both reviewers' and reviewees' learning performance, and promote students' creativity (Boud, Cohen, & Sampson, 1999; Xiao & Lucking, 2008).

Topping and Ehly (2001) believed that peer assessment is one of the most important ways of peer-assisted learning, providing students with a way to learn from each other, which is beneficial for developing knowledge and skills. While students are evaluating their peers' work, they can imitate the strengths of the work, a view which happens to coincide with that of Schunk (1987), while having students with similar abilities and background learn together and imitate each other brings positive learning effects. Tsivitanidou, Zacharia, and Hovardas (2011) discovered that peer assessment learning activities can encourage positive learning attitudes on the part of students. Tseng and Tsai (2010) further pointed out that using online peer assessment has the advantage of being anonymous, and students do not need to worry that they will be identified while giving scores, which encourages them to raise doubts and propose the advantages and drawbacks of others' work to achieve a better learning effect. Davies (2000) used a computer system to administer a peer assessment activity, and it was found that after comparing their work with that of others, the students had higher self-awareness and demands, which has a positive meaning for education. Hwang, Hung, and Chen (2014) adopted a quasi-experiment in natural science courses in which the elementary school students were asked to design games; the results showed that with the peer assessment mechanism, the students' learning achievement, learning motivation, and problem-solving ability were effectively increased. Khonbi and Sadeghi (2012) found that students learning with peer assessment performed better than those with self-assessment.

To date, several studies have confirmed that peer assessment in most learning activities is an effective and reliable learning approach (Falchikov & Magin, 1997; Khonbi & Sadeghi, 2012). Students can ponder the advantages and weaknesses of peers' work before submitting their feedback; such a process provides students with a powerful chance to reflect and to assist their peers in producing work of better quality (Tsai & Chuang, 2013; Cheng, Liang, & Tsai, 2015). It also provides the opportunity to improve their critical thinking, which refers to the cognitive process, a logistical conclusion or solution gained from the judgement of learning goals (Glassner, Weinstock, & Neuman, 2005; van Gelder, 2005).

As students in peer assessment activities have a chance to reflect on their own work and that of their peers using the rubrics provided by the teacher, their critical thinking awareness would be increased, especially when using online peer assessment tools (Joordens, Pare, & Pruesse, 2009; Lynch, McNamara, & Seery, 2012). In peer assessment activities, students can review their learning process and their results by evaluating others' work and accepting others' feedback, helping them to find their own learning blind spots and reconstruct their learning goals and plans (Cheong & Cheung, 2008). On the other hand, many researchers have further pointed out that using computer technology to learn can cultivate students' critical thinking awareness (Dwyer, Hogan, & Stewart, 2014; Wang, Huang, & Hwang, 2016). Therefore, in this study, an online peer-assessment strategy is employed in a Scratch programming activity.

Online peer-assessment system for computer programming

To evaluate the effectiveness of the proposed teaching strategy, an online peer assessment system was developed and provided for the experimental group students to conduct peer assessment, as shown in Figure 1. There are three databases in the peer-assessment environment: (1) a "work" database for students to save their Scratch programming works, student personal information, and peer assessment database; (2) a "personal information" database to keep the students' profiles, such as their names, student IDs and contact information; and (3) a "peer-assessment" database to save the ratings and comments provided by the students and teachers.

In addition, there are several function units that assist students and teachers during the peer-assessment process: (1) a "Do project" function to enable students to upload their programming projects; (2) a peer-assessment system to support assessment activities by providing an anonymous matching mechanism (i.e., the function unit that automatically assigns students' work to their peers), an instant reminder mechanism (i.e., the function unit to inform the students to review peers' work), and an instant feedback mechanism (i.e., the function unit to inform the students to browse the ratings and comments from peers); and (3) a "Monitor system" for teachers to supervise the peer-assessment process.

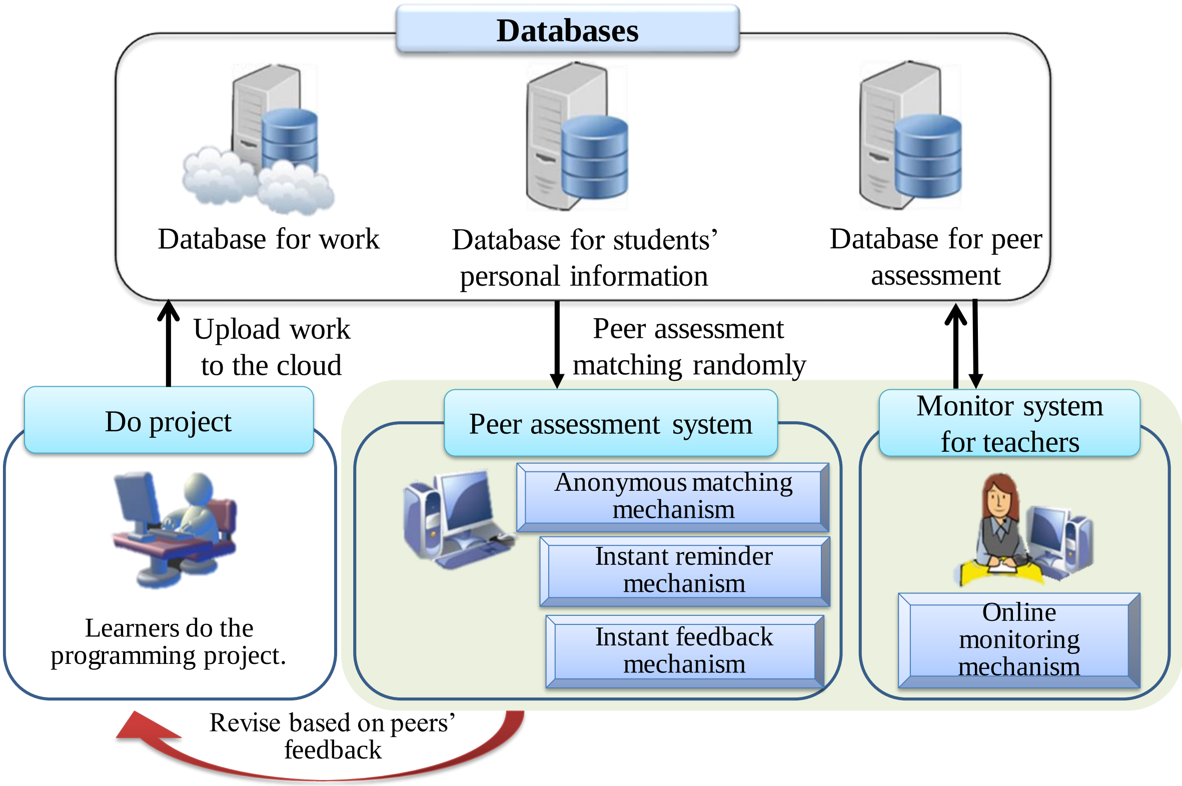

Figure 1. Online peer assessment system for computer programming projects

Figure 2 shows the user interface of the peer-assessment system. When reviewing peers' work on a computer screen, the students can rate the work and provide comments on the other screen.

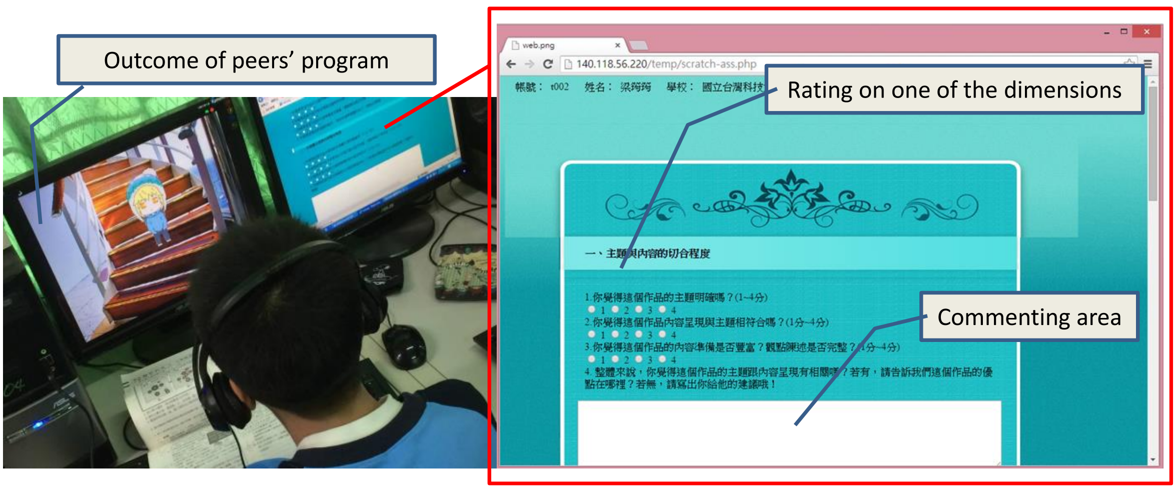

Figure 2. Screenshot of peer assessment in the online peer assessment system

When the students submit their scores and feedback, a reminder window is displayed for the reviewers to check their comments. Besides, another anonymous comment is given once the reviewer submits his/her comments, as shown in Figure 3. Furthermore, the system also offers the teacher the function to monitor the status of peer assessment.

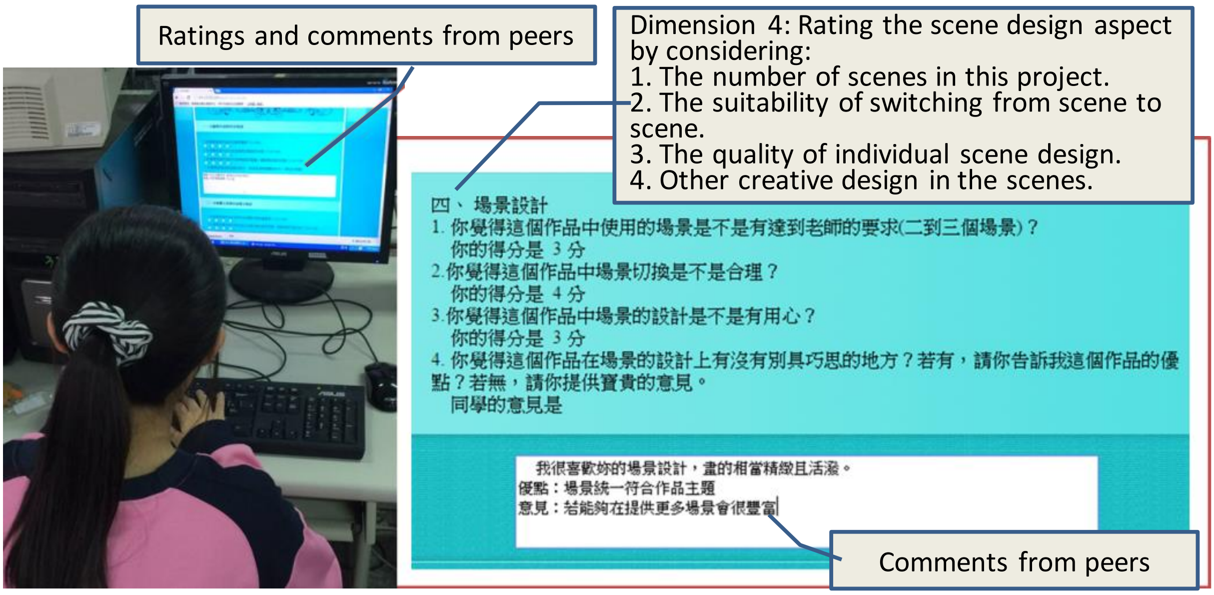

Figure 3. Review peers' comments on their work

Experimental design

Participants

The participants of this study were from four classes of 166 ninth graders in a junior high school located in southern Taiwan who learned Scratch programming in their computer courses. A quasi-experimental design was adopted, and two classes of 80 students were assigned to the experimental group, learning with the online anonymous peer assessment teaching strategy. Every student gave scores and comments for the work assigned by the system. The other two classes of 86 students were assigned to the control group, learning with the traditional teacher feedback teaching strategy. The teaching activity was led by an experienced teacher with over 30 years of teaching experience in information technology. After the activity, the students were asked to make a Scratch programming on the topic of school life or on a social issue of their choice.

Experimental procedure

The experiment took 10 weeks of two hours per week, as shown in Figure 4. In the first week, the students took a prior knowledge test about the Scratch programming language and completed the pre-questionnaires of learning attitude and critical thinking. In weeks 2-5, the teacher taught how to program for four weeks, and the students were asked to develop a program as the programming skills pre-test. Following that, one class period was spent on the introduction of the programming design project and the rubrics of the work.

In weeks 6 and 7, the students did the programming project. In weeks 8 and 9, the students in the experimental group were situated in the online peer assessment activity, including evaluating peers' work based on the rubrics provided by the teacher (see Table 1) and revising their work after receiving ratings and comments from peers. On the other hand, those in the control group received comments from the teachers and revised their work accordingly. It should be noted that the lowest rating in the rubrics was "1" rather than "0" to avoid overly discouraging the reviewees. Similar rating schemes have been adopted by several previous studies (Hsia et al., 2016; Hwang, Hung, & Chen, 2014).

In order to examine the effectiveness of the proposed teaching strategy, in the tenth week, all of the students were asked to finish a Scratch program as the post-test of programming skills; following that, they took the programming knowledge post-test and completed the questionnaires of learning attitude and critical thinking awareness.

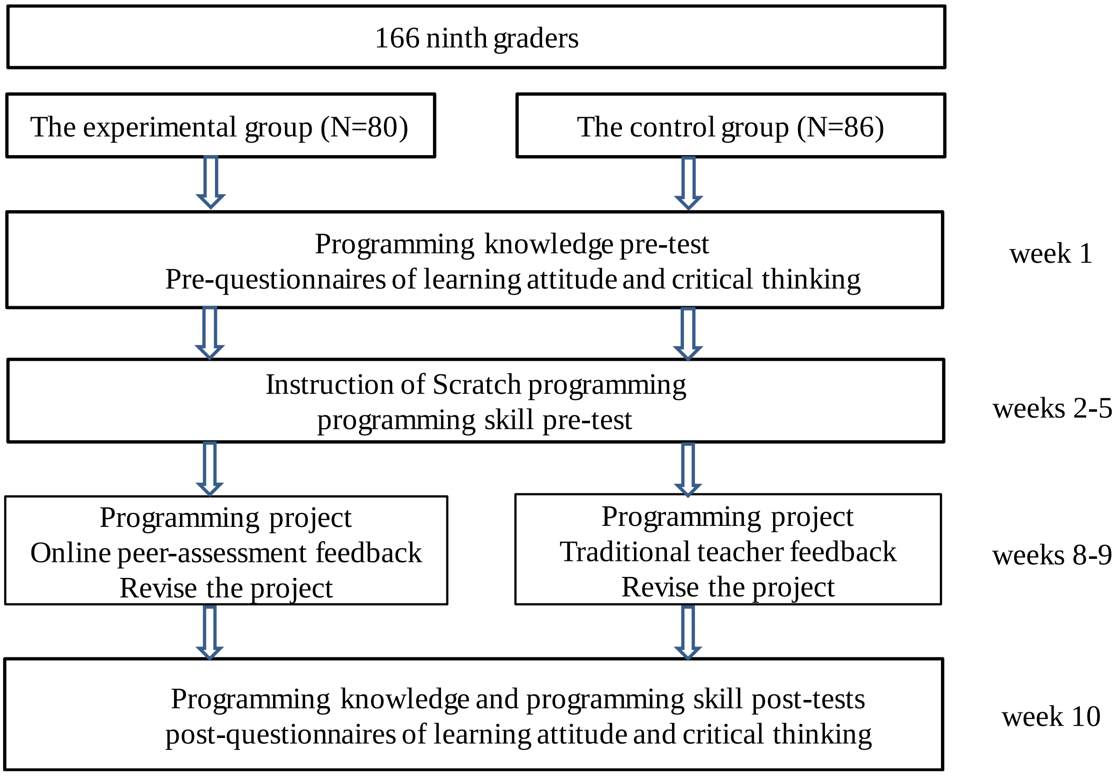

Figure 4. Experimental process

Table 1. Rubrics of online peer assessment

Dimension	4 points (Excellent)	3 points (Good)	2 points (Fair)	1 point (Poor)
Relevant level of topics and contents	The content and topic are highly relevant. There is bountiful content with detailed descriptions.	Most of the content is relevant to the topic but it can be more bountiful.	Little content in the project is relevant to the topic and there should be more content.	The content and the topic are totally irrelevant.
Level of integration of multimedia elements and content	High integration of pictures, videos, and contexts.	Most of the integration of pictures, videos, and contexts is great.	There is little integration of pictures, videos, and contexts.	No integration of pictures, videos, and contexts.
Joyfulness and innovation of contents	The project is presented vividly with multiple characters.	Most of the content is fun and there are many characters, but only a few have special features.	Only one character is presented in the project, but the character has special features.	The content is boring with only one character with no features.
Gaming scene design	There are more than three scenes in the project.	There are three scenes in the project.	There are two scenes in the project.	There is only one scene, making the game monotonous.

Measuring tools

To achieve the research goal, the measuring tools in this study included the programming knowledge tests, programming skill test, and the questionnaires of learning attitude and critical thinking.

The programming knowledge tests included a pre-test and a post-test designed by two teachers who had taught the computer technology course for over 15 years. The pre-test aimed to test the students' Scratch prior knowledge, using 15 true-or-false test items and 20 multiple-choice test items, with a perfect score of 100. The post-test aimed to test the students' knowledge of Scratch programming, such as the meaning and syntax of variables, array, conditional branch, loop statements, input, output and the control operations, using 10 true-or-false test items, 15 multiple-choice test items, and 5 fill-in-the-blank test items, with a perfect score of 100. The KR20 values of the programming knowledge pre-test and post-test were 0.71 and 0.73, respectively. As both the values were greater than 0.7, it is concluded that the reliability of the tests was good according to the suggestions of Nunnally (1978).

The programming skill pre-test and post-test were two programming projects, which aimed to evaluate the students' competence of using the programming statements and operations to develop Scratch programs based on the topics specified by the teacher. The program was evaluated by two teachers based on the rubrics in Table 1. The consistency degrees (i.e., Pearson correlation coefficients) of the ratings given by the two teachers were 0.95 and 0.92 for the programming skill pre-test and post-test, respectively.

The learning attitude questionnaire was developed by Hwang, Yang, and Wang (2013). There were nine items and the Cronbach's alpha value was .79 in this study, showing that the reliability of the questionnaire was acceptable.

The critical thinking awareness questionnaire was revised based on the measure developed by Dwyer, Hogan and Stewart (2014). It aimed to measure the students' tendency to review and reflect on their learning status and the problems they were facing, such as "I periodically examine if I have achieved the learning objectives" and "When facing a problem, I would question if there are better choices or solutions." The Cronbach's alpha value of the questionnaire was .85 in this study, showing that the questionnaire was of good reliability.

Results

Analysis of programming knowledge and skills

The programming knowledge pre-test aimed to test if the students had equivalent programming background knowledge, while the post-test aimed to evaluate their programming knowledge after the learning activity. On the other hand, the Scratch programming pre-test was used to assess the students' programming skills before the learning activity, while the Scratch programming post-test was used to evaluate their programming skills after the activity. The Scratch programs developed by the students were assessed by two experienced teachers to ensure the objectives of the evaluation.

A test for the homogeneity of regression on the programming knowledge pre-test scores was conducted and it was found that there was no significant difference between the scores of programming knowledge ($F = 0.818$, $p = .367$) and programming skills ($F = 3.038$, $p = .083$) of the two groups, meaning that they had equivalent programming knowledge and skills before the activity.

Table 2. ANCOVA result of the learning achievement post-test and programming skills of the two groups

Variance	Group	*N*	Mean	*SD*	Adjusted mean	Std. error	*F*
Knowledge test	Experimental	80	74.1	10.5	73.8	1.37	29.92***
	Control	86	62.9	14.8	63.3	1.32	
Scratch project	Experimental	80	83.57	10.38	83.23	0.47	35.83***
	Control	86	73.38	10.94	73.70	0.64	

Note. $^{***}p < .001$.

An analysis of covariance (ANCOVA) was then conducted on the students' post-test scores of programming knowledge and skills by adopting their pre-test scores of programming knowledge and skills as the covariate variables, respectively. As shown in Table 2, the experimental group, with online peer assessment teaching strategy outperformed the control group, with traditional teacher feedback. For the programming knowledge test, there was a significant difference between the scores of the two groups ($F = 29.92$, $p < .001$), and the experimental group had a higher score mean. Moreover, for the Scratch program, the experimental group also performed better than the control group ($F = 35.83$, $p < .001$). Such results indicate that the online peer-assessment approach brought better learning effects on students' programming knowledge and skills than the conventional teaching.

Analysis of critical thinking awareness

To explore the effects of the proposed teaching strategy on students' critical thinking awareness, ANCOVA was used to analyze the students' post-questionnaire ratings by excluding the impacts of their pre-questionnaire ratings. A test for the homogeneity of regression on the pre-questionnaire of critical thinking awareness was conducted, and no significant difference was found between the two groups of students for critical thinking awareness ($F = 1.90$, $p > .05$). An analysis of covariance was performed on the post-questionnaire scores, and the results showed that with different teaching strategies, a significant difference was reached for critical thinking awareness ($F = 21.23$, $p < .001$), as shown in Table 3. These results indicate that an online peer assessment approach can reinforce students' critical thinking awareness.

Table 3. ANCOVA result of the critical thinking awareness post-questionnaire scores of the two groups

Variance	Group	*N*	Mean	*SD*	Adjusted mean	Std. error	*F*
Critical thinking awareness	Experimental	80	3.78	0.67	3.80	0.07	21.23^{***}
	Control	86	3.38	0.67	3.36	0.07	

Note. $^{***}p < .001$.

Analysis of learning attitudes

To explore the effects of the proposed teaching strategy on the students' learning attitudes, ANCOVA was used to analyze the students' post-questionnaire ratings by excluding the impacts of their pre-questionnaire ratings. A test for the homogeneity of regression on the pre-questionnaire of learning attitude was conducted, and no significant difference was found between the two groups of students for learning attitude ($F = 3.20$, $p > 0.05$), meaning that the two groups of students had similar levels of learning attitude. An analysis of covariance was performed on the post-questionnaire scores, and the results showed that with different teaching strategies, a significant difference was reached for the students' learning attitudes ($F = 19.53$, $p < 0.001$), as shown in Table 4.

Table 4. ANCOVA result of the learning attitude post-questionnaire scores of the two groups

Variance	Group	*N*	Mean	*SD*	Adjusted mean	Std. error	*F*
Learning attitude	Experimental	80	3.84	0.67	3.88	0.06	19.53^{***}
	Control	86	3.52	0.73	3.49	0.06	

Note. $^{***}p < .001$.

Discussion and conclusions

In this study, an online peer-assessment approach for computer programming activities was proposed, and a peer-assessment system was developed accordingly. A learning activity was conducted in a high school Scratch programming course to evaluate the effectiveness of the proposed approach. After analyzing the experimental data, the researchers of this study found that the students participating in the programming projects with the online peer assessment approach outperformed those participating in the projects with the conventional teaching and practicing approach in terms of programing knowledge, programming skills, critical thinking awareness, and learning attitude.

As indicated by a number of researchers, students face several challenges when learning computer programming knowledge and skills (Sáez-López, Román-González, & Vázquez-Cano, 2016). One challenge is the difficulty for novices to realize the meaning and syntax of a programming language which is very different from those of the natural language they use. Another challenge is to foster students' competence of developing computer programs based on the programming knowledge they have learned. The adoption of visualized programming environments such as Scratch might solve part of these problems since students do not need to write programming statements using a language they are not familiar with. Instead, they can focus on understanding the meaning of the programming instructions and on learning to express their programming logic. In the meantime, the lead-in of peer-assessment could further guide them to realize the criteria of developing a computer program and to reflect on their programming project based on their observations of peers' work as well as the feedback from their peers (Khonbi & Sadeghi, 2012; Tseng & Tsai, 2010; Vanderhoven, Raes, Schellens, & Montrieus, 2012).

Several previous peer-assessment studies have reported that deep involvement in learning projects by playing the role of a reviewer is able to encourage students to learn better (Hsia, Huang, & Hwang, 2016; Xiao & Lucking, 2008). When playing the role of a reviewer, students are guided to think from the teacher's point of view by referring to the rubrics provided by the teacher. Such a process is helpful to them for realizing the criteria of a successful work, as well as encouraging them to elaborate their work, which could promote their knowledge and skills of completing the learning tasks (Boud, Cohen, & Sampson, 1999; Lai & Hwang, 2015). Researchers have also indicated that the feature of anonymity in online peer assessment can encourage students to provide more comments on peers' work with less hesitation (Gielen, Peeters, Dochy, Onghena, & Struyven, 2010).

As indicated by Wang, Hwang, and Hwang (2016), the peer-assessment activity is a process of critical thinking. Students need to state the advantages and weaknesses of peers' work and their own after playing the role of a reviewer. This is beneficial to them in terms of promoting their critical thinking awareness. In the online peer-assessment environment, the students were able to ponder on and revise their comments before sending any messages to their peers; that is, the online peer-assessment environment provided opportunities for the students to elaborate their comments and evaluate their peers' work by thinking critically and deeply, confirming what was reported in the study of Joordens, Pare, and Pruesse (2009).

Consequently, in the Scratch programming environment with the peer-assessment approach, students are likely to have deep involvement, see things from different perspectives, and be more willing to provide comments, which could also be the reasons why they showed better programming knowledge and skills, learning attitude toward the programming course, and awareness of critical thinking than those who learned in the Scratch programming environment without the peer-assessment approach.

There is, however, one limitation of this study; that is, the findings might not be able to be inferred to those computer programming activities that use traditional programming environments with text interfaces instead of visualized environments like Scratch. This implies that further studies are required to investigate the effects of online peer assessment on students' programming performance in different programming environments.

Acknowledgements

This study is supported in part by the Ministry of Science and Technology of the Republic of China under contract numbers NSC 102-2511-S-011 -007 -MY3 and MOST 104-2511-S-011-001-MY2.

References

Barr, V., & Guzdial, M. (2015). Advice on teaching CS, and the learnability of programming languages. *Communications of the ACM, 58*(3), 8-9.

Boud, D., Cohen, R., & Sampson, J. (1999). Peer learning and assessment. *Assessment & Evaluation in Higher Education, 24*(4), 413-426.

Brito, M. A., & de Sá-Soares, F. (2014). Assessment frequency in introductory computer programming disciplines. *Computers in Human Behavior, 30*, 623-628.

Chang, K. E., Wu, L. J., Weng, S. E., & Sung, Y. T. (2012). Embedding game-based problem-solving phase into problem-posing system for mathematics learning. *Computer & Education, 58*(2), 775-786.

Cheng, K. H., Liang, J. C., & Tsai, C. C. (2015). Examining the role of feedback messages in undergraduate students' writing performance during an online peer assessment activity. *The Internet and Higher Education, 25*, 78-84.

Cheong, C. M., & Cheung, W. S. (2008). Online discussion and critical thinking skills: A Case study in a Singapore secondary school. *Australasian Journal of Educational Technology, 24*(5), 556-573.

Davies, P. (2000). Computerized peer assessment. *Innovations in Education and Training International, 37*(4), 346-355.

Dwyer, C. P., Hogan, M. J., & Stewart, I. (2014). An Integrated critical thinking framework for the 21st century. *Thinking Skills and Creativity, 12*, 43-52.

Falchikov, N. (2004). Involving students in assessment. *Psychology Learning and Teaching, 3*(2), 102-108.

Falchikov, N., & Magin, D. (1997). Detecting gender bias in peer marking of students' group process work. *Assessment & Evaluation in Higher Education, 22*(4), 385-396.

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

Gielen, S., Peeters, E., Dochy, F., Onghena, P., & Struyven, K. (2010). Improving the effectiveness of peer feedback for learning. *Learning and Instruction, 20*(4), 304-315.

Glassner, A., Weinstock, M., & Neuman, Y. (2005). Pupils' evaluation and generation of evidence and explanation in argumentation. *British Journal of Educational Psychology, 75*, 105-118.

Hovardas, T., Tsivitanidou, O. E., & Zacharia, Z. C. (2014). Peer versus expert feedback: An Investigation of the quality of peer feedback among secondary school students. *Computers & Education, 71*, 133-152.

Hsia, L. H., Huang, I., & Hwang, G. J. (2016). Effects of different online peer-feedback approaches on students' performance skills, motivation and self-efficacy in a dance course. *Computers & Education, 96*, 55-71.

Hwang, G. J., Hung, C. M., & Chen, N. S. (2014). Improving learning achievements, motivations and problem-solving skills through a peer assessment-based game development approach. *Educational Technology Research and Development*, *62*(2), 129-145.

Hwang, G. J., Yang, L. H., & Wang, S. Y. (2013). A Concept map-embedded educational computer game for improving students' learning performance in natural science courses. *Computers & Education, 69*, 121-130.

Joordens, S., Pare, D. E., & Pruesse, K. (2009). peerScholar: An Evidence-based online peer assessment tool supporting critical thinking and clear communication. In *Proceedings of the 2009 International Conference on e-Learning* (pp. 236–240). Academic Conferences Limited.

Kátai, Z. (2015). The Challenge of promoting algorithmic thinking of both sciences-and humanities-oriented learners. *Journal of Computer Assisted Learning, 31*(4), 287-299.

Keppens, J., & Hay, D. (2008). Concept map assessment for teaching computer programming. *Computer Science Education, 18*(1), 31-42.

Khonbi, Z. A., & Sadeghi, K. (2012). The Effect of assessment type (self vs. peer) on Iranian university EFL students' course achievement. *Procedia - Social and Behavioral Sciences 70*, 1552 – 1564.

Kim, S. H., & Song, K. S. (2012). The Effects of thinking style based cooperative learning on group creativity. *Creative Education, 3,* 20-24.

Kobsiripat, W. (2015). Effects of the media to promote the scratch programming capabilities creativity of elementary school students. *Procedia - Social and Behavioral Sciences, 174*, 227-232.

Krpan, D., Mladenović, S., & Rosić, M. (2015). Undergraduate programming courses, students' perception and success. *Procedia-Social and Behavioral Sciences, 174*, 3868-3872.

Lai, C. L., & Hwang, G. J. (2015). An Interactive peer-assessment criteria development approach to improving students' art design performance using handheld devices. *Computers & Education, 85*, 149-159.

Liu, N.-F., & Carless, D. (2006). Peer feedback: The Learning element of peer assessment. *Teaching in Higher Education, 11*(3), 279-290.

Lynch, R., McNamara, P. M., & Seery, N. (2012). Promoting deep learning in a teacher education programme through self- and peer-assessment and feedback. *European Journal of Teacher Education*, *35*(2), 179-197.

Moreno, J. (2012). Digital competition game to improve programming skills. *Educational Technology & Society*, *15* (3), 288–297.

Nunnally, J. C. (1978). *Psychometric theory* (2nd ed.). New York, NY: McGraw-Hill.

Ouahbi, I., Kaddari, F., Darhmaoui, H., Elachqar, A., & Lahmine, S. (2015). Learning basic programming concepts by creating games with scratch programming environment. *Procedia - Social and Behavioral Sciences 191*, 1479 -1482.

Peppler, K., & Kafai, Y. (2005). *Creative coding: The Role of art and programming in the K-12 educational context*. Cambridge, MA: MIT Media Laboratory.

Peppler, K., & Kafai, Y. (2007). From SuperGoo to Scratch: Exploring media creative production in an informal learning environment. *Journal on Learning, Media, and Technology, 32*(2), 149-166.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM, 52*(11), 60–67.

Sáez-López, J. M., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A Two year case study using "Scratch" in five schools. *Computers & Education, 97*, 129-141.

Schunk, D. H. (1987). Peer models and children's behavioral change. *Review of educational* research, *57*(2), 149-174.

Topping, K. (1998). Peer assessment between students in colleges and universities. *Review of Educational Research, 68*(3), 249-276.

Topping, K. (2009). Peer assessment. *Theory into Practice, 48*(1), 20-27.

Topping, K. J., & Ehly, S. E. (2001). Peer-assisted learning. *Journal of Educational and Psychological Consultation, 12*(2), 113-132.

Tsai, Y. C., & Chuang, M. T. (2013). Fostering revision of argumentative writing through structured peer assessment. *Perceptual & Motor Skills, 116*(1), 210–221.

Tseng, S. C., & Tsai, C. C. (2010). Taiwan college students' self-efficacy and motivation of learning in online peer assessment environments. *The Internet and Higher Education*, *13*(3), 164-169.

Tsivitanidou, O. E., Zacharia, Z. C., & Hovardas, T. (2011). Investigating secondary school students' unmediated peer assessment skills. *Learning and Instruction, 21*, 506-519.

Van Gelder, T. (2005). Teaching critical thinking: Some lessons from cognitive science. *College Teaching, 53*, 41-46.

Vanderhoven, E., Raes, A. Schellens, T., & Montrieus, H. (2012). Face-to face peer assessment in secondary education: does anonymity matter? *Procedia - Social and Behavioral Sciences, 69,* 1340-1347.

Wang, H. -Y., Huang, I., & Hwang, G.-J. (2016). Comparison of the effects of project-based computer programming activities between mathematics-gifted students and average students. *Journal of Computers in Education, 3*(1), 33-45.

Williams, L., Wiebe, E., Yang, K., Ferzli, M., & Miller, C. (2002). In Support of pair programming in the introductory computer science course. *Computer Science Education, 12*(3), 197-212.

Xiao, Y., & Lucking, R. (2008). The Impact of two types of peer assessment on students' performance and satisfaction within a Wiki environment. *The Internet and Higher Education*, *11*(3), 186-193.

Yang, T.-C., Hwang, G.-J., Yang, S. J. H., & Hwang, G.-H. (2015). A Two-tier test-based approach to improving students' computer-programming skills in a web-based learning environment. *Educational Technology & Society, 18*(1), 198-210.