Enhancing the programming skill in high school engineering education via flipped classroom and peer assessment

Ren-Hung Hwang[1], Jang-Jiin Wu, Zheng-Yu Tsai, Pao-Ta Yu, Chin-Feng Lai
Professor, Ph.D. candidate, Ph.D. candidate, Professor, Associated Professor
Department of Computer Science and Information Engineering
National Chung Cheng University
Chia-Yi, 621 Taiwan, ROC
E-mail: rhhwang@cs.ccu.edu.tw, jangjiin@gmail.com, tcy97p@cs.ccu.edu.tw, csipty@cs.ccu.edu.tw, cinfon@cs.ccu.edu.tw

Keywords: programming skill, flipped classroom, peer assessment

1 INTRODUCTION

The new generation of engineering students has grown up with innovative technologies, such as the Internet, smart phone environment, and applications, which affect how they learn. One is now starting to re-think how to teach students in this new generation, how they can gain knowledge through videos before their classroom instruction, and how instructors can guide their students to clarify and apply that knowledge during their lesson in a flipped classroom. Recently, many events have been aimed at promoting computer programming education for K-12 students worldwide. For example, American President Barack Obama stated in a video for the 2013 Computer Science Education Week annual event: “Don’t just buy a new video game - make one. Don’t just download the latest app - help design it. Don’t just play on your phone - program it.” British Education Minister Michael Gove believes that if you want the United Kingdom to appear like IT companies, it needs to pay more attention to childhood education programs, and therefore academic institutions in the UK, as well as Google and Microsoft, and other IT companies, had formally requested in September, 2014, that students from five to 16 years of age must learn computer program design in schools. When students are 14 years old, teachers will guide them on how to use two or more programming languages. The British government wants their young learners to not only consume technology, but also to build it — instead of just playing computer games, they might be able to create them one day. In Taiwan, this problem has also been addressed, with many teachers promoting a free software Scratch education program to strengthen education, either during their lessons or out-of-school time.

[1] Corresponding Author
Ren-Hung Hwang
rhhwang@cs.ccu.edu.tw

Scratch is a popular tool for teaching K-12 students computer programming which has been developed by the Massachusetts Institute of Technology (MIT). With a visual development platform, Scratch was designed for K-12 students with very little computer experience. Its visual interface helps them to just use the mouse to drag, be able to give the component action logic, and intuition can be used to design interactive stories, games, music, and other art works. In addition, the Scratch website provides a forum where students can upload their creations and share them with everyone, thereby encouraging collaborative group work.

The recent increase in popularity of massive open-access online courses (MOOCs) has made it possible for anyone with an internet connection to enrol for university level courses. However, while new web technologies allow for scalable ways to deliver video lecture content, implement social forums, and track student’s progress in MOOCs, one remains limited in one’s ability to evaluate and give feedback for complex assignments such as computer program design problems. In order to enhance student learning, this study used flipped learning strategy, and enrolled the three round peer assessment with a Scratch game programming project. How does one make sure that the students have acquired what they were supposed to learn online? In this study, online homework and surveys were conducted to gauge students’ understanding, as it helps both the student and the instructor to prepare for the in-class portion of the flipped classroom. By reviewing their peers’ work, students consolidate, reinforce, and deepen, both their own and their peers’ creativity and skill on Scratch game programming. It also helps students to build critical analytic skills and become comfortable with receiving criticism. A web-based learning system is built to conduct this study.

2 LITERATURE REVIEW

2.1 Basic concept of the flipped classroom

Recent technological developments have given rise to the flipped classroom, which is one approach to blended learning. With the popularity of computers, mobile devices, and the Internet, flipped learning has recently gained significance. A flipped class is one that inverts the typical cycle of content acquisition and application so that students can gain necessary knowledge before the class begins, and instructors then guide students to interactively clarify and apply that knowledge during the class. Teachers could then take more time to help students who are behind, and students could practice their assignment, or discuss it with their peers, to promote a higher level of thinking skills [1].

It was found that the flipped classroom approach demonstrates advantages such as access to instruction at anytime and anywhere, increase student engagement, better use of classroom time, discussions with peers and teacher, and access to expert advice from others [2][3].

Many studies have pointed out the general framework about flipped learning in higher education, but there is little evidence concerning the use of flipped learning as an approach to improve student’s performance in high school programming education. Thus, this study aims to demonstrate the effectiveness of teaching computer programming in high schools using a flipped learning approach.

2.2 Why teach programming

One can learn powerful problem-solving/design/thinking strategies by learning to program. This is because, when students program, they first need to find a solution to a problem, and then they need to reflect on how to communicate their solution to the computer, following the syntax and grammar of a programming language, through an exact way of how a computer thinks [4][5]. Since the ability to program is an essential part of engineering, students are presented with challenging programming problems in a number of courses.

Programming is seen as a skill to communicate, in an unambiguous way, a set of instructions to an unintelligent computer [6]. If this process could take place by means of a relatively simple programming language, (e.g., Scratch), offering a simpler syntax than other commonly used programming languages, students could focus more on the semantic aspect of the program and produce fewer syntax errors [7]. The process should then lead students to first write simple programs, and then to combine the simple solutions together, thereby obtaining solutions to many more complicated problems [8].

2.3 The effects of using peer assessment

Numerous studies have perceived the usefulness of web-based peer assessment [9], and students were observed learning effectively from reading peers' work and comments [10]. A web-based learning system often provides activities that allow students to interact with each other, such as peer-assessed homework, and class forums. For a class with large number of students, peer assessment offers a solution that can scale the grading of complex assignments [11].

In many flipped classes, peer assessment was conducted for only one round. However, it has been found that many students had improved their project over three rounds in an on-line peer assessment study [12, 13]. Therefore, the three-round peer assessment model was adapted for this study. Furthermore, most of the peer-assessment research studies were implemented in global higher education; study on the usage of online peer assessment in high school level has received less attention. This study has designed online peer assessment activities in a high school engineering course, and scales both evaluation and peer learning in creative domains.

3 METHOD

3.1 Participants

The participants were 111 students from three different classes in a computer course at a vocational senior high school in central Taiwan. Every student enrolled in the computer course was required to design a Scratch game project by team work with three or four members. The duration of the study was an 18-week period with two sessions for each week. Students were required to learn Scratch programming, game design, programming creation, and peer assessment on the Web-based learning system. To learn Scratch programming skills, the students had to engage in the following activities, i.e. watching the video materials about Scratch programming in a web-based learning system, giving online reflection about the material, discussions in class, etc. A total of 33 individual interesting projects were generated. The students were asked to comment on the other two group's projects anonymously through a three-round peer assessment on the web-based learning system. Then they were asked to revise their own group's project by taking five peers' comments and suggestions.

3.2 Procedure of experiment

Week 1 (preparation): The teacher began the course with a brief overview of the programming introduction. After introducing the Web-based online learning and assessment system, students become familiar with the system's functions.

Week 2-5 (flipped learning): The students learned about the Scratch programming by watching the video teaching materials, and then gave the five degree feedback: fully understanding, reasonable understanding, partial understanding, limited understanding, and non-understanding. After the feedback was received, the teacher was then able to design easier or harder teaching materials.

Week 6-9 (game programming project): Each group in this study was asked to design a Scratch game programming project.

Week 10-16 (three-round peer assessment): Students submitted their game project first, and then experienced three rounds of peer assessment, and then had two chances of modifying their project. Two teachers assessed all of the students' projects. The students were asked to comment on the other two group's projects, and they were asked to revise their own group's project by taking their peers' comments and suggestions.

Table 1 showed the procedure of this experiment.

Table 1. Procedure of experiment

Week	Subject	Task
Week 1	Teacher	1. Provide instructions on how to watch online materials. 2. Explain scoring criteria and provide students with sample peer assessments.
	Students	1. Get basic understandings about Scratch programming. 2. Learn how to perform peer-assessment.
Week 2-5	Teacher	1. To deliver the online materials. 2. Discuss with students in the classroom.
	Students	1. Watch the online materials. 2. Discuss the materials with teacher and classmates.
Week 6-7	Teacher	1. Deliver the Scratch programming project. 2. Discuss with student about their game project plan.
	Students	1. Form group of three or four members. 2. Develop game programming project.
Week 8-9	Teacher	1. Monitor students' project creation.
	Students	1. Develop Scratch programming project with team members. 2. Have preliminary and final versions of projects submitted.
Week 10	Teacher	1. Monitor students' project submission status. 2. Give a reminder for late work. 3. Grade the students' projects.
	Students	1. Revise and upload projects. 2. Grade other team's projects.
Week 11-12	Teacher Students	See week 8-9
Week 13	Teacher Students	See week 10
Week 14-15	Teacher Students	See week 8-9
Week 16	Teacher Students	See week 10
Week 17-18	Teacher	Gather and analyze the data, figure out the students' final score.
	Students	Oral presentation about the project in the classroom.

4 RESULTS AND DISCCUSSION

4.1 Did the Senior high school students improve the quality of their projects in a Scratch programming course after the three-round peer assessment activity?

Paired *t*-tests were used to compare students' peer assessment scores, and teacher's scores changed through the three rounds, as shown in *Table 2*. It was found that students progressed during their performance. Students significantly improved their game projects by the peer rounds and peer assessment activities.

Table 2. Paired t tests on-line peer assessment students' score and teachers' score changes through three rounds. (n=33)

	Peer Evaluation Rounds			Teacher Evaluation Rounds		
	1st->2nd	2nd->3rd	1st->3rd	1st->2nd	2nd->3rd	1st->3rd
Creativity	-2.853***	-3.019***	-7.007***	-4.927***	-2.602***	-7.266***
Multimedia	-4.252***	-3.559***	-9.616***	-5.288***	-4.028***	-8.14***
Completeness	-4.49***	-3.877***	-7.199***	-4.49***	-4.403***	-7.74***

*** p<0.001

4.2 What were the relationships between the scores made by the learning peers and those determined by teachers?

Table 3 shows the Pearson's correlation coefficient between peer scores and teacher scores on a three outcome variable. *Table 3* shows that the scores between peer and teacher were significantly highly correlated (r=0.81-0.909, p<0.01). It also shows that online peer assessment in senior high school could be perceived as a valid assessment method. This finding was consistent with the conclusion in [10].

Table 3. The Pearson's correlation between peer and teacher assessment score. (n=33)

Round	Evaluation dimension		
	Creativity	Multimedia	Completeness
First Round	0.81**	0.892**	0.822**
Second Round	0.871**	0.863**	0.895**
Third Round	0.909**	0.865**	0.824**

** p<0.01

4.3 What were the relationships between the feedbacks in which the students were given after they watched the materials, and the subsequent performance of their projects?

Table 4 demonstrated the coefficients of Pearson's correlation between peer assessment scores during the three rounds, and the students' feedback after they watched the teaching video materials. The results showed that the third round scores determined by the learning peers were significantly highly correlated with students' responses.

Table 4. The Pearson's correlation between material feedback and peer assessment

Round	Evaluation dimension		
	Creativity	Multimedia	Completeness
First Round	0.451**	0.397**	0.137
Second Round	0.589**	0.500**	0.552**
Third Round	0.713**	0.563**	0.797**

** p<0.01

Table 5 demonstrated the coefficients of Pearson's correlation between the teacher's scores during the three rounds, and the students' feedback after they watched the teaching video materials.

It was found that for each dimension, a student team group received a higher project score if the team members understood the video materials.

Table 5. The Pearson's correlation between material feedback and teacher assessment

Round	Evaluation dimension		
	Creativity	Multimedia	Completeness
First Round	0.567**	0.374**	0.18
Second Round	0.705**	0.627**	0.577**
Third Round	0.724**	0.681**	0.771**

** p<0.01

4.4 What was the relationship between the types of peer feedback in which the students obtained, and the subsequent performance of their projects?

This study classified the peer responses into corrective, reinforcing, suggestive, and didactic feedback, based on [14]. After classifying, only two didactic feedbacks were acquired in round 1, and therefore that type of feedback was discarded.

Table 6 presents the relationship between the types of peer feedback after the first round, and peer-assessment scores after the second round. The results indicated that the Reinforcing feedback of the first round was positively correlated with student performance in all dimensions (r=0.439, 0.422, 0.38, for Creativity, Multimedia, Completeness, respectively, p<0.05). Moreover, the Corrective feedback was also related to students' scores on "Creative" and "Multimedia" (p<0.05). That suggested the Reinforcing and Corrective feedback should be constructive in students' development of their work. *Table 7* shows the relationship between the types of peer feedback in which students received from the second and third round peer assessment scores they received. These three types of peers' feedback played a positive role on students' quality of programming projects in all dimensions (p<0.05). These results, in general, showed that the feedback from peers was useful in helping students to better develop their programming projects.

Table 6. The relationship between the peer feedback after the first-round and peer scores after the second-round.

Round 1, Feedback classification	Round 2, Evaluation dimension		
	Creativity	Multimedia	Completeness
Corrective feedback	0.589*	0.532*	0.368
Reinforcing feedback	0.439*	0.422*	0.38*
Suggestive feedback	0.42*	0.35	0.32

* p<0.05

Table 7. The relationship between the peer feedback after the second-round and peer scores after the third-round.

Round 2, Feedback classification	Round 3, Evaluation dimension		
	Creativity	Multimedia	Completeness
Corrective feedback	0.692**	0.423*	0.471**
Reinforcing feedback	0.464**	0.423*	0.471**
Suggestive feedback	0.441*	0.414*	0.439*

* p<0.05
** p<0.01

5 SUMMARY

This study explored the effects of online self-learning and the validity of online peer assessment in high schools, and to analyze the effect of various types of peer feedback on students. The participants consisted of 111 senior high school students enrolled in a Scratch programing course. The following procedure was performed during the class for 18 weeks: the students watched the online video materials, gave their feedback, and formed 33 groups of three or four members to create Scratch game projects, viewed other groups' work, and performed assessment on the Web-based learning system. The peer assessment activity consisted of three rounds. Pearson's correlation analysis was conducted and the results indicated: 1) students significantly improved their projects by being involved with the peer assessment activities; 2) the peer assessment results, and the teacher assessment results, were both without significant differences, indicating that peer assessment in high schools could be perceived as a valid assessment method; 3) the online video materials feedback and end-of-project final score were highly consistent. This study also examined the relationship between the types of peer feedback in which students obtained from peer assessment, and then the subsequent performance of their projects. Corrective peer feedback was useful in helping students' develop better projects.

This study was geared towards developing curricula for students to enhance their programming skills in high school engineering education via flipped classroom and peer assessment. In this study, the web-based flipped classroom learning and peer assessment system was developed for a Scratch game programming project in a senior high school computer course. After the students watched and learned basic programming skills from the online video materials, they could give material feedback. The instructor then collected their comments to check whether or not the students had understood the concepts of the materials and guided students to interactively clarify and apply that knowledge during class. This study explored the relationship between the online video materials responses and the end-of-project final score. It was found that if the students were able to more easily understand the concept of online materials, they could develop a better learning achievement. That is, as teachers provided more suitable online material, students have responded by performing better.

The three-round peer assessment process took about nine weeks to improve the programming skills on this project. The students had an opportunity to observe their peers projects through the assessment learning process, while learning their peers programming skills. Students modified their original work to achieve a better quality, constructing and refining their programming skills through peer interactions.

It was found that the students in the present study improved their projects by these three round peer assessment exercises, the peer assessment results, and the teacher assessment results, were without significant differences, indicating that peer assessment in high school could be perceived as a valid assessment method.

This study also explored the relationship between the types of peer feedback and the students' subsequent performance on the programming project. It was found that Reinforcing feedback is helpful to promote the quality of student projects. Corrective feedback and Suggestive feedback also helped in the later stage of the peer assessment process.

According to our findings, peer and teacher assessment results were significant and consistent, implying that the Web-based project peer assessment was adequate to reflect learning achievements. Therefore, the Web-based project peer assessment could be regarded as a valid method. When a course is opened and has many students, such as MOOCs, peer-assessment will be more than useful in high schools, and therefore should be well worth further study that includes future research

REFERENCES

[1] Khan, S. (2012), The one world schoolhouse: Education reimagined London, Hodder and Stoughton.

[2] Fulton, K.P. (2012), 10 Reasons to Flip. New Styles of Instruction, Vol.94, No.2, pp. 20-24.

[3] Milman, N. (2012), The flipped classroom strategy: What is it and how can it best be used? , Distance Learning, Vol.9, No.3, pp. 85-87.

[4] Saeli, M., Perrenet J., Jochems W. M.G., and Zwaneveld B. (2011), Teaching Programming in Secondary School: A Pedagogical Content Knowledge Perspective. Informatics in Education, Vol. 10, No. 1, 73–88.

[5] Szlávi, P., and Zsakó, L. (2006), Programming versus application. In: Mittermeir, R.T. (Ed.), ISSEP 2006, LNCS4226, pp. 48–58.

[6] Hromkovic, J. (2006), Contributing to general education by teaching informatics. In: Mittermeir, R.T. (Ed.), ISSEP 2006, LNCS, 4226, pp. 25–37.

[7] Mannila, L., Peltom¨aki, M., and Salakoski, T. (2006), What about a simple language? Analyzing the difficulties in learning to program. Computer Science Education, Vol. 16, No. 3, pp. 211–227.

[8] Abelson, H., and Sussman, G.J. (1996), Structure and Interpretation of Computer Programs, 2nd edition. Series MIT Electrical Engineering and Computer Science.

[9] Tsai, C.-C., Lin, S. S. J., and Yuan, S. M. (2002), Developing science activities through a networked peer assessment system. Computers & Education, Vol. 38, pp. 241 – 252.

[10] Tseng, S.-C., and Tsai, C.-C. (2007), On-line peer assessment and the role of the peer feedback: A study of high school computer course, Computers & Education, Vol. 49, pp. 1161-1174.

[11] Piech, C., Huang J., Chen Z., Do C., Ng A., and Koller D. (2013), Tuned Models of Peer Assessment in MOOCs, EDM 2013, pp. 153-160.

[12] Lin, S. S. J., Liu, E. Z. F., and Yuan, S. M. (2001), Web-based peer assessment: feedback for students with various thinking-styles. Journal of Computer Assisted Learning, Vol. 17, pp. 420 – 432.

[13] Tsai, C.-C., Lin, S. S. J., and Yuan, S. M. (2002), Developing science activities through a networked peer assessment system. Computers &Education, Vol. 38, pp. 241 – 252.

[14] Chi, M. T. H. (1996), Constructing self-explanations and scaffolded explanations in tutoring. Applied Cognitive Psychology, Vol. 10, pp. 33 – 49.