Development of a Peer-Interaction Programming Learning System

Pham-Duc Tho[1,2(✉)], Nguyen-Hung Cuong[1,2], Hoang-Cong Kien[1], and Chih-Hung Lai[2]

[1] Hung Vuong University, Viet Tri, Vietnam
thopham@hvu.edu.vn
[2] National DongHwa University, Hualien County, Taiwan

Abstract. Computer programming is basic knowledge in the digital age and becoming an critical subject during recent years. However, learning to programme is not an easy topic as supported by many researchers. During the development of information technology, many online learning systems have been developed and proven their positive effect on students learning. However, few studies have geared toward supporting its use in programming courses with peer-interaction. Therefore, this study aimed to develop an online learning system named Peer-Interaction Programming Learning System. The system was developed and being used by many programming classes both in Vietnam and Taiwan. In this paper, we reported on the design of the system and its user interface, discussed our motivation and underlying teaching philosophy.

Keywords: Online learning system · Programming learning · Peer interaction

1 Introduction

Programming is a complex activity with some factors that could contribute to its difficulty. Jenkins [1] pointed out that the teaching methods employed by the instructor is the primary effect to students achievement during programming courses, but Matthíasdóttir [2] argue that the problematic nature of computer programming is the actual cause. Gomes and Mendes [3] in another research argue that some of the issues contributed to programming activity are the study methods, abilities and attitudes employed by the student, also the nature of the art of programming, the lack of prior knowledge of novice students, and the psychological influence that the student suffered from society [4].

From past studies, the learning benefits of online learning systems have been well recognized [5]. Allen and Seaman [6] reported that 77.1% of academic leaders in America agreed that online learning is critical to their long-term strategy. Increasing numbers of institutions have offered online courses to accommodate students' needs and also to reduce their budget [7]; however, online programming courses have been problematic for many students [8]. Students lack motivation and low self-efficacy for learning and may lead to lower completion rates than in face-to-face courses [9]. One of the possible reason is the lack of peer interaction and less immediate feedback from the instructor [10, 11].

P. Cong Vinh and V. Alagar (Eds.): ICCASA 2018/ICTCC 2018, LNICST 266, pp. 163–170, 2019.
https://doi.org/10.1007/978-3-030-06152-4_14

Little empirical research has been conducted in remedial online programming courses with regard to peer interaction. Law, Lee and Yu [12] suggest that social pressure and competition have a significant and positive relationship with efficacy during their research with online programming learning system named Programming Assignment aSsessment System (PASS). For decades, researchers have been building online learning systems to lower the barrier to programming learning [2, 9, 12–15]. For example, CodeWrite and StudySieve are systems created by Denny, Luxton-Reilly, Tempero and Hendrickx [16] which is aimed at helping students learning programming by using question posing and peer comment. Both CodeWrite and StudySieve have a significant effect in support students learn programming [16, 17] but only focus on free-response domain which limits students who wanted to create other types of questions. Also, only after the solution compiles and passes all the test cases are the solutions submitted by other students revealed [16] makes ordinary students harder to solve the intricate questions without any hints or supports. Although their research is helpful in improving students' achievement in programming courses, more empirical research is necessary for peer-interaction settings due to the explosion of social network nowadays. Therefore, this study aimed to develop a programming learning system named Peer-Interaction Programming Learning System (PIPLS) to fulfill the gaps which remain in previous research.

2 Description of the System

PIPLS is a system first developed in 2015 in National Donghwa University, Taiwan, with the primary aim to assisting beginners in learning programming with the Student Question Generation (SQG) strategy [9]. It is now regularly used as an integrated part of many undergraduate courses related to computer programming.

PIPLS is designed aiming at fill some gaps which are remains from previous systems:

- Allow students to choose to use their real name, their nickname or anonymous.
- Support more question types: multiple choice, short answer, true-false, fill-in-the-blank, coding and essay with an automatic judge or semi-automatic which is significant benefit for staff.
- Make the peer-interactive process more accurate and accessible for students whom nowadays familiar with many social networking sites.
- Allow learning content to be integrated into courses.

PIPLS supports student-generated multiple types of questions, included free-response, multiple-choices, fill in the blanks, and true-false questions. In this system, the students can discuss with each other by asking and answering the questions. The teachers can set questions, share the resources of learning and develop the effectiveness of class management.

After logging in, via the home page (Fig. 1), students can find some quick statistic information about their progress: courses they are following, contributed questions, answered questions, unanswered questions, and exercises with grade.

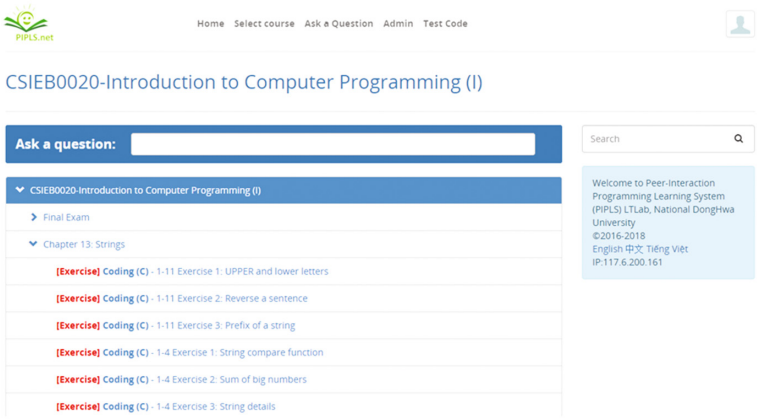

Fig. 1. The main page of PIPLS

Besides those sections, there are many features and functions that we developed to help students and teachers. The system was developed on the basis of the Question2Answer system, so it has inherited all features of Question2Answer [18]. Due to the limit of this paper, the role of some of these sections is described next.

2.1 Composing a Question

This section (Fig. 2) allows a student to compose a question. The student was asked to choose the type of question before they can reach this section. The student can post open question to Discussion board or post regular question as an exercise to their classmates. PIPLS is designed to support multiple question types, including free-response, multiple-choices, fill in the blanks, and true-false questions.

Inherited from Question2Answer [18], the question title must be provided with detailed information and embed multimedia, links, … To help students find and organize relevant questions, the questions may be tagged with appropriate topics by the author.

Original Question2Answer only provided free-response question so we developed additional fields applied for different types of question. Figure 2 illustrates how a coding question is defined.

Besides important information come with types of question, we also developed available time for the question, anonymous feature, mark question as exercise specifically for teacher, and the maximum answer per user allowed for the question.

2.2 Answering Questions and Peer Interactions

In PIPLS, answers are revealed according to course setting. Teacher can allow student to view others answer by default, after deadline or only after the student submitted the correct answer for automatically judged questions, or the answer for an essay question.

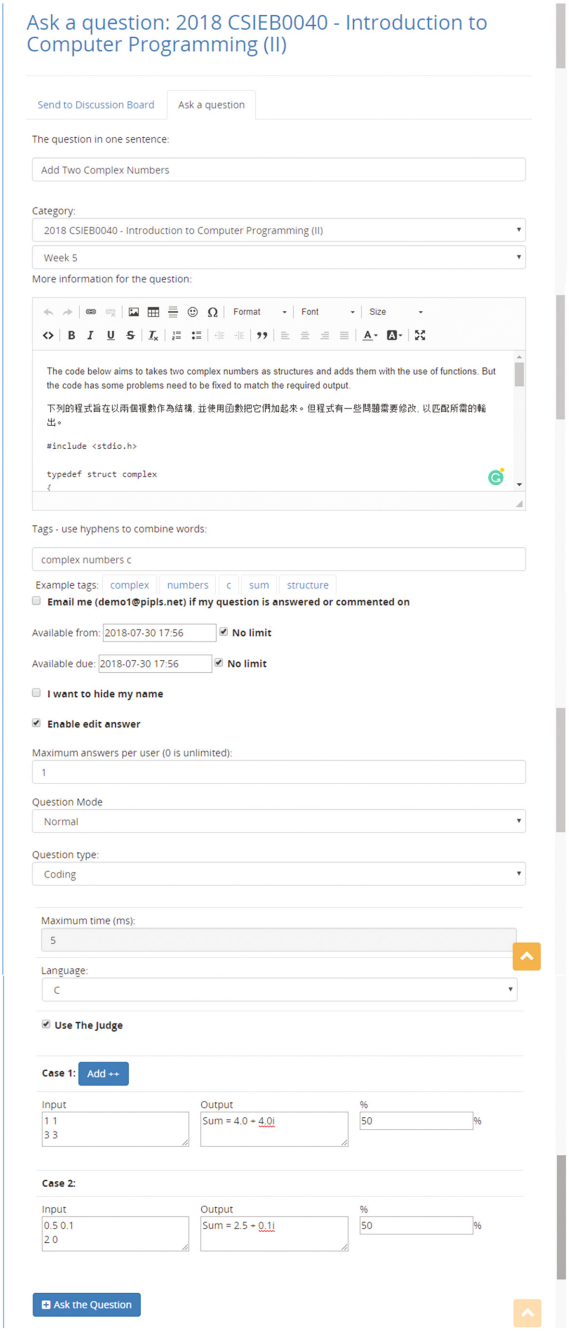

Fig. 2. Composing a new question in PIPLS

We enhanced PIPLS by including many functions of traditional online learning systems, include “call for help” function. When student cannot figure out the answer, they don’t need to give up or require help by giving some comment and wait. They can keep thinking straight without losing time by “call for help”. This function will allow student reading the answers from other classmates without knowing which answer is correct. Then the student need to decide which answer is correct and complete their own answer.

According to peer-interaction features, students also have the opportunity to write formative feedback to the question author, thanks to the comment feature of original Question2Answer which is visible to all users, and can agree or disagree with other feedback provided by their peers by voting feature. When the others’ answer is visible, students can give comment and also voting in others’ answer (Fig. 3).

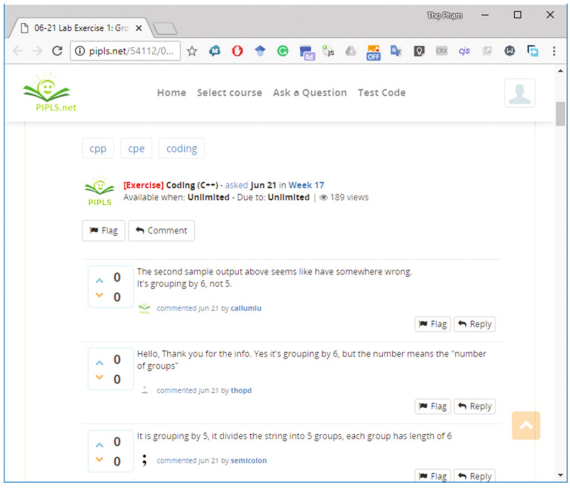

Fig. 3. Students and TAs comments to a question

All the notifications in PIPLS will be sent to students by Facebook Messenger, students also able to post question to discussion board or reply to another’s comment via Facebook Messenger (Fig. 4).

2.3 Evaluating Questions and Answer

PIPLS support multiple-choice, true/false, fill in the blanks, and coding questions so it can automatically generate feedback for students who answer questions, by reporting whether the answer is correct (by percentage) or not. The student needs to submit the correct answer in order to see other answers. And numbers of answer are limited by author (or not) in the question composing interface (Fig. 5).

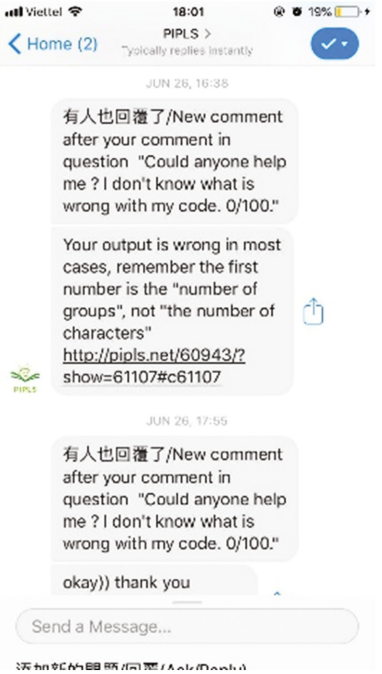

Fig. 4. Facebook Messenger notification

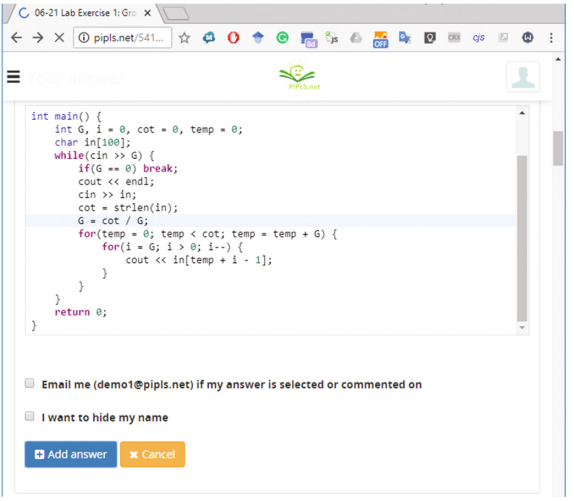

Fig. 5. Answer a Coding question in PIPLS

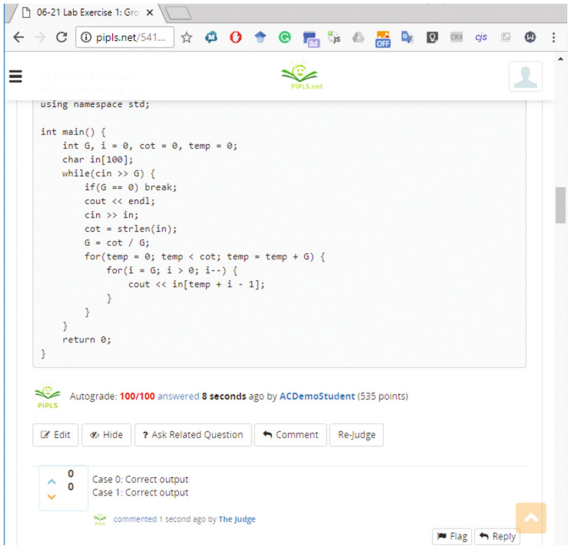

Fig. 6. Immediate feedback after student submit answer

In the PIPLS, we have two types of free-response question: essay questions and coding question. Essay questions needs author or teacher to examine but other type of questions are automatically judged. PIPLS now supporting C, C++, Java, Pascal, Python, JavaScript and PHP in auto-judge function (Fig. 6).

Coding questions are not only judged automatically, teacher also can re-judge the answer in case the machine cannot or if teacher want to give some bonus points for the good solution.

3 Conclusion and Future Work

We developed the system based on previous research and focused on supporting students to learn Programming. We also extended the type of question generation (multiple-choice, true/false, fill in the blanks, essay, and coding) and developed many additional features. We hope to give more support to students when compared with other systems which also support programming learning.

In the future, we will plan to enhance the existed systems' functions and evaluate the impact of the tool on students' performance. We also intend to study the nature and quality of the artifacts (questions, answers, and feedback) produced by students.

References

1. Jenkins, T.: On the Difficulty of Learning to Program. Citeseer (2002)
2. Matthíasdóttir, Á.: How to teach programming languages to novice students? Lecturing or not. In: International Conference on Computer Systems and Technologies-CompSysTech (2006)
3. Gomes, A., Mendes, A.J.: Learning to program-difficulties and solutions (2007)
4. Jenkins, T.: On the difficulty of learning to program. In: Proceedings of the 3rd Annual HEA Conference for the ICS Learning and Teaching Support Network, pp. 1–8 (2002)
5. Gökçearslan, Ş., Alper, A.: The effect of locus of control on learners' sense of community and academic success in the context of online learning communities. Int. Higher Educ. **27**, 64–73 (2015)
6. Allen, I.E., Seaman, J.: Online Report Card: Tracking Online Education in the United States. Babson Survey Research Group (2016)
7. El Said, G.R.: Understanding how learners use massive open online courses and why they drop out: thematic analysis of an interview study in a developing country. J. Educ. Comput. Res. **55**, 724–752 (2017)
8. Dolgopolovas, V., Jevsikova, T., Dagiene, V.: From Android games to coding in C—An approach to motivate novice engineering students to learn programming: a case study. Comput. Appl. Eng. Educ. **26**, 75–90 (2018)
9. Lai, C.H., Tho, P.D., Liang, J.S.: Design and evaluation of question-generated programming learning system. In: 2017 6th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI), pp. 573–578 (2017)
10. Echeverría, L., Cobos, R., Machuca, L., Claros, I.: Using collaborative learning scenarios to teach programming to non-CS majors. Comput. Appl. Eng. Educ. **25**, 719–731 (2017)
11. Lu, O.H.T., Huang, J.C.H., Huang, A.Y.Q., Yang, S.J.H.: Applying learning analytics for improving students engagement and learning outcomes in an MOOCs enabled collaborative programming course. Interact. Learn. Environ. **25**, 220–234 (2017)
12. Law, K.M.Y., Lee, V.C.S., Yu, Y.T.: Learning motivation in e-learning facilitated computer programming courses. Comput. Educ. **55**, 218–228 (2010)
13. Thomas, L., Ratcliffe, M., Woodbury, J., Jarman, E.: Learning styles and performance in the introductory programming sequence. SIGCSE Bull. **34**, 33–37 (2002)
14. Bergin, S., Reilly, R.: Programming: factors that influence success. SIGCSE Bull. **37**, 411–415 (2005)
15. Funabiki, N., Korenaga, Y., Nakanishi, T., Watanabe, K.: An extension of fill-in-the-blank problem function in Java programming learning assistant system. In: 2013 IEEE Region 10 Humanitarian Technology Conference (R10-HTC), pp. 85–90 (2013)
16. Denny, P., Luxton-Reilly, A., Tempero, E., Hendrickx, J.: CodeWrite: supporting student-driven practice of java. In: Proceedings of the 42nd ACM Technical Symposium on Computer Science Education, pp. 471–476. ACM, Dallas (2011)
17. Luxton-Reilly, A., Denny, P., Plimmer, B., Bertinshaw, D.: Supporting student-generated free-response questions. In: Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education, pp. 153–157. ACM, Darmstadt (2011)
18. Greenspan, G., Contributors: Question2Answer (2016)