Designing a Scaffolding Inquiry-based instruction to Promote Non-engineering students in STEM Learning

Chia-Jung CHANG*
Department of Information Communication, Yuan Ze University, Taiwan
* ccj@saturn.yzu.edu.tw

Abstract: The purpose of this study is to design a scaffolding inquiry-based instruction to facilitate non-engineering students in STEM learning and investigates its impact on students' computer programming, science learning and problem-solving performance. Employing a pretest-posttest experimental design, the investigation involved 19 university students who participated in a 3-week scaffolding inquiry-based program. The results showed that the instruction have positive effects on non-engineering students' logical thinking, debug, and control in computer programming and concept, high-level cognition, practice, apply and communication in science learning. Although the instruction improved students problem-solving performance, some students failed to solve complicated problems.

Keywords: scaffolding, programming, non-engineering, STEM

1. Introduction

Numerous research and educators have paid much attention on STEM education for nurturing students' various competencies. Within STEM activities, students are frequently required to design products or propose solutions to solve problems by integrating different disciplines. The process facilities students' conceptual understanding and problem analysis by the experience of applying and integrating knowledge and skills from disciplines (Dasgupta, Magana, & Vieira, 2019). Hence, STEM has been extensively implemented in varied educational contexts.

Previous studies have indicated that STEM education contributes to the enhancement of problem-solving and computer programming competencies. For example, Lin, Wang, and Wu (2019) designed a modeling-based physics programming instruction for STEM learning, indicating that the modelling process not only enhanced students' programming abilities and physics concepts, but also helped them program solutions to solve problems. Additionally, other studies indicated the experiencing in STEM activities could fosters students' problem-solving competency (Chen et al., 2020).

Recently, a growing body of research has shifted its focus towards non-engineering students within STEM learning. (Hu, Yeh, & Chen, 2020; Lin, Yu, Shih, & Wu, 2021). For instance, Hu et al. (2020) investigated the effect of hands-on activity for non-engineering students on STEM learning. They found that the activity enhanced students' learning performance and attitude by experiencing the problem-solving process. Another similar study by Lin et al. (2021) explored the AI literacy of non-engineering students after participating in STEM-based AI hands-on activity. The results revealed that STEM hands-on activity can improve non-engineering students' AI literacy. However, it is relatively challenging for non-engineering students who lacked learning experience in STEM-related subjects in the past. Previous studies indicated that non-engineering students have a lower level of engineering skills and computer literacy than engineering students (Mansoor & Ahmad, 2019). Moreover, non-engineering students often fail to perform STEM activities, which may produce a negative learning attitude toward STEM learning (Lo et al., 2017). In other words, STEM learning does not always guarantee effective learning for non-engineering students (Lin et al., 2021). Hence, how to help non-engineering students with STEM learning is an important educational issue.

Although STEM problem-solving activities benefit students, some studies found potential learning problems during students try to solve problems. Tan et al. (2023) indicated that most students encounter problems related to the application of disciplinary knowledge and practice when taking part in STEM problem-solving activities. More specifically, students often applied low-level scientific concepts to solve problems. The same problem is also found in the study by Chang and Tao (2021), indicating that even with inquiry-based instruction, students still failed to use high-level conceptions of learning science in STEM activity. Therefore, they suggested that an intended connection to reduce the gap between science and practice is helpful for students in generating solutions and enhancing STEM competencies.

To help non-engineering students engage in STEM learning, the study developed a scaffolding inquiry-based instruction in STEM learning and investigated the impact of the instruction on non-engineering students' performance in computer programming, science learning, and problem-solving. More specifically, research questions are presented as follows.

1. Does the scaffolding inquiry-based instruction facilitate non-engineering students' computer programming self-efficacy?
2. Does the scaffolding inquiry-based instruction improve non-engineering students' science learning self-efficacy?
3. What are non-engineering students' problem-solving performance in the scaffolding inquiry-based STEM?

2. Method

2.1 Participants

A pilot study using a pretest-posttest experimental design was conducted to explore the effect of the scaffolding inquiry-based instruction on STEM learning. The participants of this study were 19 university non-engineering students who were designs backgrounds (16 females and 3 males). They have no experience in making a game and taking part in STEM education. Hence, these students are suitable as subjects.

2.2 The scaffolding inquiry-based learning activity

The study designed a game-making learning activity based on an inquiry-based process to guide students on the procedural of game development using programming and provided scaffolding questions to help students establish a connection between physics and computer programming. The activity was created on CoSci (https://cosci.tw), developed by Prof. Liu, and used an inquiry process as an instructional framework to engage students in STEM learning. The activity consisted of two parts, including a training in programming and a problem-solving task. The former aimed to teach students the operation of the game design interface, basic concepts of programming, and physics concepts by a sequential inquiry process, including playing, exploration, explanation, practice, and reflection phases.

Except for the playing phases, every phase provided scaffolding questions to guide students in understanding the required tasks. For instance, during the exploration phase, students were required to observe the data sheet provided by the platform and to respond to questions regarding variables that needed to be considered for aircraft/bomb movement. At the explanation phase, the students were required to explain the game object movement using physics concepts. Moreover, students needed to explain the relationships among variables from the previous phase. These questions aimed to help students identify the role of each variable in the game and to explore the relationship among the variables by explaining physics concepts. It was expected that the support of the scaffolding question facilitated the integration and application of disciplinary knowledge. The last phase, the activity provided reflective questions to prompt students evaluate their learning performance and monitor the errors they made while designing a mini game.

Since students lacked computer programming, physics, mathematics and problem-solving skills, an aircraft shooting a tank mini game as a demonstration to help students understand the concept of programming and free fall motion concept. More specifically, the demonstration guided students in establishing the relationships among the variables and physics concepts, to control the movement of the game objects, to create interaction with keyboard input, and handling collision detection. Additionally, there are practice and reflection sessions that provided learning opportunities to reinforce the application and integration of physics and programming. Finally, students were required to complete a problem-solving game task. Differentiating from previous demonstration game, the task was a bombing mission where a player controlled a tank to dodges enemy aircraft attacks and simultaneously shot an enemy aircraft. In the task, there are four problems, including rendering game objects, objects controls, object movement, collision detection.

2.3 Procedure

Before the activity, two questionnaires, including computer programming self-efficacy scale (CPSES) and science learning self-efficacy (SLSE), were used for 30 minutes as a pretest. The scaffolding inquiry-based learning activity lasted for nine 50-min sessions over three weeks. The training was implemented in six sessions and the task took three sessions. After the activity, the same two questionnaires were employed as a posttest.

2.4 Data collection and analysis

The computer programming self-efficacy scale developed by Tsai, Wang, and Hsu (2018) was used to measure students' perception of programming learning before and after the activity. Additionally, the study used the science learning self-efficacy scale (Lin & Tsai, 2013) to investigate their perceptions of science learning. The two questionnaires were slightly revised to align with the learning activity context. The overall Cronbach's alpha of the two revised questionnaires were .93 and .94, respectively, indicating that the questionnaires were sufficiently reliable. The responses to the questionnaires were analyzed by a paired t-test to understand the changes before and after the activity. Besides, students' performance in the task was evaluated. The task consisted of four problems, each worth 4 points, for a total of 16 points.

3. Results

Table 1 showed the result of students' computer programming self-efficacy scale using a paired t-test. The results indicated that compared with before the activity, students perceived a higher level of computer programming skills, including the dimensions of logical thinking ($t=-3.16$, $p=.005<0.01$), debug ($t=-3.13$, $p=0.006<.01$), and control ($t=-3.50$, $p<.01$). However, the algorithm was no statistically significant difference between before and after the activity. In other words, students perceived a similar level of algorithm before and after the activity. The results suggested that the activity may be helpful for non-engineering students in condition statements, revised programming problems and control of programming interface.

Table 1. *The result of students' computer programming self-efficacy before and after the scaffolding problem-based activity*

Dimensions	Pretest		Posttest		t-value	p-value
	Mean	SD	Mean	SD		
Logical thinking	3.66	0.94	4.72	0.95	-3.16**	0.005
Algorithm	2.98	0.69	3.44	1.29	-1.65	0.117

Debug	3.46	0.75	4.35	1.09	-3.13**	0.006
Control	3.47	1.11	4.61	1.15	-3.50**	0.003

*<.05, **<.01, ***<.001

The result of students' science learning self-efficacy using a paired t-test was shown in Table 2. Significant differences were found in all the dimensions of science learning self-efficacy. More specifically, compared before the activity, students believed that they obtained a higher level of understanding of science concept (t=-3.48, p<.01), and could explain a scientific phenomenon using science theory (t=-3.21, p<.01). In addition, they knew how to set up variables and collect data during executing game (t=-3.98, t<.01) and were able to apply science in the game design (t=-7.36, p<.00). Moreover, they were more capable of expressing their ideas properly and clearly using science (t=-8.53, p<.00). These results suggested that the activity have a positive effect on science learning.

Table 2. *The result of students' science learning self-efficacy before and after the scaffolding problem-based activity*

Dimensions	Pretest		Posttest		t-value	p-value
	Mean	SD	Mean	SD		
Conceptual understanding	2.93	0.83	3.33	0.67	-3.48**	0.003
Higher-order cognitive skills	2.87	0.87	3.65	0.56	-3.21**	0.005
Practical work	3.18	0.93	4.05	0.66	-3.98**	0.001
Application	3.06	0.68	3.74	0.46	-7.36***	0.000
Science communication	2.67	0.77	4.29	0.54	-8.53***	0.000

*<.05, **<.01, ***<.001

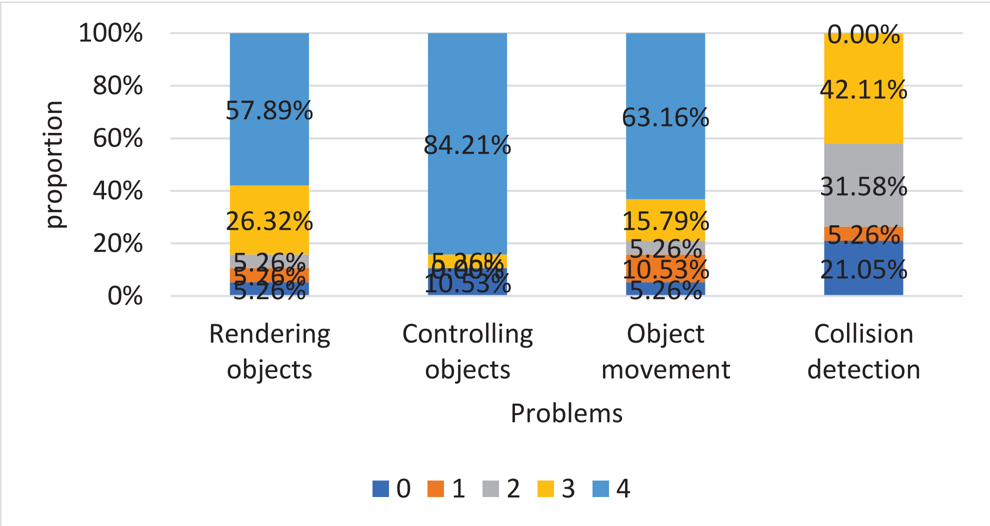

Figure 1. The results of students' problem-solving performance

Figure 1 showed the outcomes of problem-solving performance that students completed in the task. More than 80% of students could successfully render game objects in appropriated positions on the game scene and controlled objects using input devices. Approximately 75% of the students could correctly display object movement, while the rest of the students only displayed a partial correct movement and failed to move any objects (15%). Regarding collision detection, no students fully completed the four types of collision detection. Approximately 40% of the students partially achieved the task, while the rest only achieved one or two types of collision detections.

4. Conclusion

The study designed scaffolding inquiry-based learning instruction for facilitating non-engineering students in STEM learning. The results showed that the instruction contributed not only to enhancing students' computer programming, but also improving their science learning self-efficacy. However, it is worth noting that most students had difficulties in integrating programming and physics. The findings of the study serve as references for researchers and instructors in integrating scaffolding inquiry learning into STEM learning. However, there were some limitations in this study. Due to the small sample size in the study, the results may not be generalized. Another limitation was the obtained results based on the questionnaire and programming outcome. Moreover, the activity is a short-term program that lasted for three weeks. In future studies, it is suggested to verify this study with a large sample size and a long-term program, and to evaluate students' programming process to identify the potential learning difficulties.

Acknowledgements

This research was partially funded by the National Science and Technology Council, Taiwan under contract numbers MOST 110-2511-H-155-002- and MOE Teaching Practice Research Program (PEE1123014).

References

Chang, C.-J., & Tao, S.-Y. (2021). The Impact of Inquiry-based Integrated STEM on Student's Perception of Learning Science and Computer Programming. International Conference on Computers in Education.

Chen, J. C., Huang, Y., Lin, K. Y., Chang, Y. S., Lin, H. C., Lin, C. Y., & Hsiao, H. S. (2020). Developing a hands‐on activity using virtual reality to help students learn by doing. *Journal of Computer Assisted Learning, 36*(1), 46-60.

Delahunty, T., Seery, N., & Lynch, R. (2020). Exploring problem conceptualization and performance in STEM problem solving contexts. *Instructional Science*, *48*(4), 395-425.

Ge, X., & Land, S. M. (2003). Scaffolding students' problem-solving processes in an ill-structured task using question prompts and peer interactions. *Educational technology research and development, 51*(1), 21-38.

Hu, C. C., Yeh, H. C., & Chen, N. S. (2020). Enhancing STEM competence by making electronic musical pencil for non-engineering students. *Computers & Education*, *150*, 103840.

Lin, C. H., Yu, C. C., Shih, P. K., & Wu, L. Y. (2021). STEM based artificial intelligence learning in general education for non-engineering undergraduate students. *Educational Technology & Society*, *24*(3), 224-237.

Lin, T. J., & Tsai, C. C. (2013). A MULTI-DIMENSIONAL INSTRUMENT FOR EVALUATING TAIWANESE HIGH SCHOOL STUDENTS'SCIENCE LEARNING SELF-EFFICACY IN RELATION TO THEIR APPROACHES TO LEARNING SCIENCE. International Journal of Science and Mathematics Education, 11, 1275-1301.

Lin, Y. T., Wang, M. T., & Wu, C. C. (2019). Design and implementation of interdisciplinary STEM instruction: Teaching programming by computational physics. *The Asia-Pacific Education Researcher, 28*, 77-91.

Lo, K. W., Lau, C. K., Chan, S. C., & Ngai, G. (2017, October). When non-engineering students work on an international service-learning engineering project—A case study. In *2017 IEEE Global Humanitarian Technology Conference (GHTC)* (pp. 1-7). IEEE.

Mansoor, T., & Ahmad, M. S. (2019). Learning for Youth Leadership: the Case of Hong Kong, Daniel TL Shek. Grace Ngai & Stephen CF Chan, Service (Eds.), Springer: Singapore, 2019, ISBN: 978–981–13-0447-7.

Tan, A. L., Ong, Y. S., Ng, Y. S., & Tan, J. H. J. (2023). STEM problem solving: Inquiry, concepts, and reasoning. *Science & Education*, *32*(2), 381-397.

Tsai, M. J., Wang, C. Y., & Hsu, P. F. (2019). Developing the computer programming self-efficacy scale for computer literacy education. *Journal of Educational Computing Research, 56*(8), 1345-1360.