Chang, M. et al. (Eds.) (2019). Proceedings of the 27th International Conference on Computers in Education. Taiwan: Asia-Pacific Society for Computers in Education

Design Robot-Programming Activities to Engage students in the Computational Problem Solving Process

Chun-Ping Wu[a*], Jia-jyun Chen[b] & Shih-chung Li[b]
[a]*Department of Education, National University of Tainan, Taiwan*
[b]*Department of Educational Technology, TamKang University, Taiwan*
*cpwu303@gmail.com

Abstract: This study aims to design a series of robot programming learning activities to engage learners in the computational problem-solving process. The activities were designed based on the strategy of problem-based learning. To reduce any extraneous cognitive load suggested by prior research, the worked-out examples were developed and provided. Additionally, the strategy of concept-mapping was used in three different ways to facilitate students' learning. A pre-and-post quasi-experiment was conducted to explore the effects of the concept-mapping strategy. In total of 75 5th graders from three classes participated in this study for two weeks and the classes were randomly assigned to one of the three intervention conditions. The ANCOVA result supported the significant effect of gender interacting with the intervention on learners' perceived meaningfulness of programming. Implication for future studies were provided.

Keywords: Computational problem-solving, concept-maps, robot programming

1. Introduction

Computational thinking competency is critical. Individual has to develop computational thinking, making use of information technology to support deep learning, to put creative ideas into practice and to construct schema in order to solve various types of problems encountered in daily life and career. Programming education is critical to develop computational thinking because programming language is the tool we use to communicate with information technology and put our thinking into practice. Rather than focusing on "coding", learning specific languages, programming education should facilitate learners in analyzing important problems, proposing solutions and designing the commands for the technology to execute in order to test the proposed solutions (Duncan & Bell, 2014). Therefore, programming education emphasizes making use of syntactic knowledge, conceptual knowledge and strategic knowledge (Bayman & Mayer, 1988) to solve problems. Prior empirical studies supported that the problem-based learning strategy (name PBL, here-after) engaged students in practicing problem-solving, leading to schema construction (Hmelo & Evensen, 2000). Therefore, this study aimed to develop a series of robot-programming learning activities, which are based the problem-based learning strategy, to engage student in computational problem-solving process.

2. Literature review

The PBL centers on the real-world problems, which learners may encounter outside school settings (Hmelo & Evensen, 2000). During the learning process, learners may have the opportunity to play with the given problems and try to apply the newly learned contents to solve the problems. Meanwhile they may observe how to put the learned concepts into practice, which may further enhance their motivation (van Merriënboer, 2007; van Merriënboer & Kester, 2007; van Merriënboer, Kirschner, & Kester, 2003). Similarly, learners, while engaged in PBL activity, are given the opportunity to learn how to make use of programming knowledge to solve computational problems. Furthermore, PBL emphasizes the process of problems-solving and creating solutions. Learners may learn to know "what, why, how and when to use the learned concepts". This process does not only enable learners to master the learned contents, but also help them to far transfer knowledge into solving more complex problems in the future. More importantly, PBL engages learners in creating solutions. Learners, while engaged in playing with computational problems, may become technology creator, experiencing their technology empowerment during the problem-solving process and observing the meaningfulness of learning with technology.

However, the prior research studies indicated that novice learners, while interacting with the ill-structured problems, usually pay too much attention to the details of the problems, ignoring the important concepts associated with the problem (Corbalan, Kester, & van Merriënboer, 2009). This study designed a series of robot programming learning activities for the elementary school students. Robot programming is new to majority of them. To reduce any extraneous cognitive loads resulted from the activities, worked-out examples were developed in this study to help learners concentrate on critical information. Specifically the worked-out examples included critical reasoning points to approach and observe the problems and diverse aspects to think of solutions.

To enhance learners' interaction with the learned contents, the concept-mapping strategy was also proposed to guide learners to think of important concepts associated with the given problems. According to the cognitive load theory, the instructional strategy may also result in germane cognitive loads. Learners may benefit from the instructional strategy only when the cognitive load brought by the learning activity does not exceed their cognitive capacity (Sweller, 1988). In this study, the intrinsic cognitive load the learners experience may similar and the work-out example may reduce extraneous cognitive load. This study would test whether different ways to use the concept-mapping strategy may result in different levels of germane cognitive loads, which further influence the learners' learning from the computational problem-solving activities. Specifically, the concept map is used either as a presentation tool or as a knowledge representation tool. When it is used as a presentation tool by the instructor, the learners could clearly visualize the important concepts and the knowledge structure. When it is used by learners as knowledge representation tool, learners could recall important ideas or concepts they identified during the learning process. However, novice learners, when they approach a new problem, may experience higher cognitive loads in identifying the concepts and knowledge structure. Therefore, this study proposed another approach, incomplete concept-mapping with some concepts identified in the map, which may reduce the germane cognitive load to avoid exceeding cognitive capacity.

To sum up, this study developed a series of robot programming learning activities and the effect of the concept mapping was explored taking into account of gender difference.

3. Method

3.1 Participants

75 5th grade students from three classes at one elementary school in Taiwan were volunteered to participate in this study. All the participating students have parents' consent forms.

3.2 Research Design

This study included two stages. First, the course, including six units of robot programming learning, was developed using the ADDIE approach. Each unit started with one problem and students worked through the problems in a team of two persons. A learning system were developed to provide worked-out examples to assist students in the problem-solving process. Additionally, the strategy of concept-mapping was used in three different ways (i.e. the teacher used the concept maps as advanced organizer and summary tools; the students listed and map the concepts by themselves from the blank paper, and the students filled in critical concepts in the given concept-map). The lessons plans, learning systems and concept-maps were reviewed by four experts, two professors in the research area of instructional technology and two elementary school teachers who taught kids programming.

Second, a pre-and-post quasi-experimental research was conducted to examine the effects of the scaffolds. Specifically, this study developed three interventions according to the above-mentioned three different ways to use the concept-mapping strategy. The three participating classes were randomly assigned to one of the interventions. All the participants were taught by the same teachers, using the same learning contents, activities, and learning systems. Due to the time restriction of the participating school, the course was conducted only for two weeks. One week before the course started, the pre-test, including the pre-assessment and perceived problem-solving inventory was implemented. The post-test, including the post-assessment and perceived meaningfulness of programming, was conducted at the end of the study.

3.3 Instrument

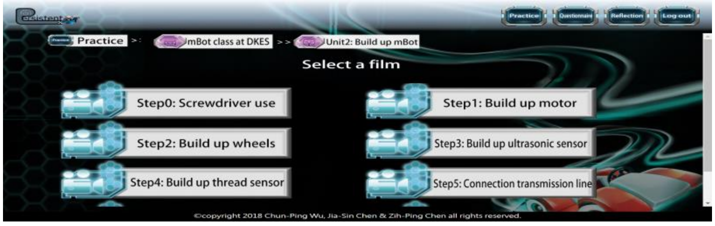
To collect participants' performance in the robot programming, two measurements were developed by the first author. The pre-assessment, including 10 multiple-choice questions, measured participants' knowledge of SCRATCH. The averaged difficulty of the pre-assessment is 0.52 while the averaged item discrimination is 0.65. The post-test, including 10 multiple-choice questions, was developed to measure participants' learning in the robot programming after this study. The averaged difficulty of the pre-assessment is 0.6 while the averaged item discrimination is 0.6.

Additionally, since students' perceived problem-solving may also influence their participation in the problem solving activities, this variable was measured by the Problem-solving inventory developed by Heppner and Petersen (1982). The authors translated the measurement into Mandarin and the translated instrument was reviewed by three experts. This instrument was pilot- tested by 85 6th grader students. The inventory include 32 items using 6 point likert scale. The higher score indicates more positive attitude and behavior toward the problem-solving process (Heppner & Petersen, 1982). The reliability reported in this study is .87. Last, participants' perceived meaningfulness of programming was measured at the end of the study using the instrument developed by Kong, Chiu, and Lai (2018). The instrument, measuring meaningfulness of programming, included 4 items, originally using 5 point likert scale. The higher score indicates more positive attitude toward the meaningfulness and value of learning programming (Kong, Chiu, Lai, 2018). The authors translated the measurement into Mandarin and modified into 6 point likert scale. The translated and modified instrument was reviewed by three experts. This instrument were pilot tested by 85 6th grader students. The reliability reported in this study is .95.

4. Results and Conclusion

4.1 Participants Design of the Learning Activities and the Scaffolds

Six units of robot programming were developed and the learning system named "Persistent" was developed by the first author. Unit1 started with a video, demonstrating application of programming in the daily life and explaining why programming is important. Then learners was presented a video demonstrating what mBOT (the robot) could accomplish. The learners were guided to break the accomplishment into smaller tasks and to think of how they could help mBOT to accomplish the tasks. In Unit 2, the learners were asked to build up the mBOT. The learners observed the mBOT and tried to put the components together. They were directed to the learning system to observe the scaffolds if they need help (See Figure1). Before moving to Unit3, the learners observed the video that was presented in Unit1 and wrote down the tasks of mBOT again. In Unit 3~5, they learn how to program the mBOT to accomplish the tasks written by themselves. Learners would work with more complex tasks if they accomplish the above-mentioned tasks. In each unit, they could access the scaffolds if they need help. In Unit 6, the learners need to create one story using two mBOTs as main characters and program the mBOT to accomplish their designed tasks in the stage.



©copyright 2018 Chun-Ping Wu, Jia-Sin Chen & Zih-Ping Chen all rights reserved.

Figure 1. Scaffolds Provided in Unit2.

This study developed two kinds of scaffolds. First, the worked-out examples were presented in the system. The purposes of the examples are to guide learners to observe the problems from different perspectives, to provide hints for analyzing problems, to provide hints for proposing solutions, to provide hints for bugs-fixing and so on. All participating learners were allowed to access the learning system any time during the course. Second, the research team identified the concepts associated with the problems embedded in each learning activity and presented the concepts in the format of concept-maps. In intervention condition 1, the teacher used these maps as advanced organizers. In intervention condition 2, the learners identified important concepts and created their own concept maps to demonstrate the relationships among concepts. In intervention condition 3, the learners were given the incomplete concept maps and they need to fill in the concepts. Please refer Figure 2 for the example of the concept map used in intervention 3.

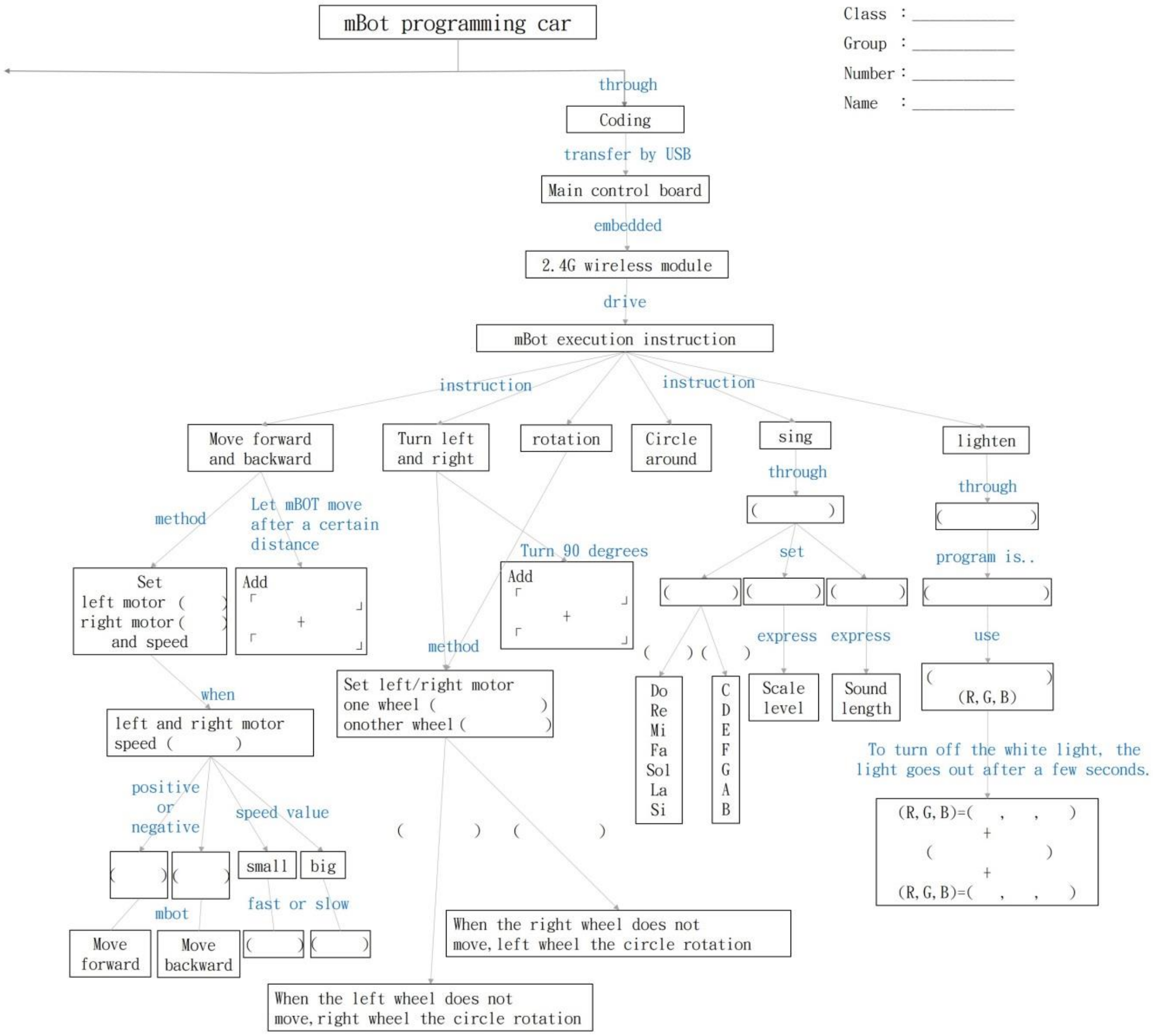


Figure 2. Example of the Concept Map Used in Intervention 3.

4.2 The Effects of the Learning Activities

The descriptive statistics of the four examined variables, including pre-assessment, post-assessment, perceived problem-solving and perceived meaningfulness of programming, are presented in Table 1. As observed, boys scored a little bit higher than girls in all variables. Further analyses, taking into consideration of gender, were conducted.

Table 1

Descriptive Statistics

		Intervention 1 (N=26; b=15/g=11)		Intervention 2 (N=23; b=12/g=11)		Intervention 3 (N=26; b=15/g=11)	
		Mean	SD	Mean	SD	Mean	SD
Pre-assess ment	Boy	5.53	2.72	5.83	2.44	5.53	2.72
	Girl	4.09	2.77	4.54	2.29	4.09	2.77
	Total	4.92	2.78	5.21	2.41	5.50	2.51
Post-assess ment	Boy	6.60	2.38	6.58	2.50	6.60	2.38
	Girl	6.09	1.57	4.81	1.66	6.09	1.57
	Total	6.38	2.06	5.73	2.28	5.84	2.79
Perceived problem-solving	Boy	4.36	0.78	4.23	0.92	4.36	0.78
	Girl	3.83	0.53	3.96	0.71	3.83	0.53
	Total	4.14	0.72	4.11	0.82	4.31	0.69
Meaningful ness	Boy	5.65	0.57	5.02	1.53	5.65	0.57
	Girl	3.77	1.94	4.65	1.65	3.77	1.94
	Total	4.85	1.60	4.84	1.56	5.05	0.98

First, using the ANCOVA to control the effects of the pre-assessment, the effect of gender interacting with the intervention on students' programming performance in the post-assessment was not supported (F=1.08, *p=.35).* Second, the scores of perceived problem-solving is significantly correlated with scores of perceived meaningfulness of programming (r=.59, *p<.001).* The ANCOVA used the scores of perceived problem-solving as the covariate. The result show that the effect of gender interacting with the intervention on students' perceived meaningfulness of programming reached statistically significance (F=3.17, *p=.05).* The interaction effect was furtherly presented in Figure3. While controlling the effect of perceived problem-solving, the boys perceived programming activities more meaningful than girls especially in the group while the teacher taught the programming using concept maps as an advanced organizer. Instead, girls, while they were asked to fill in the incomplete concept maps, perceived programming activities a little more meaningful than boys

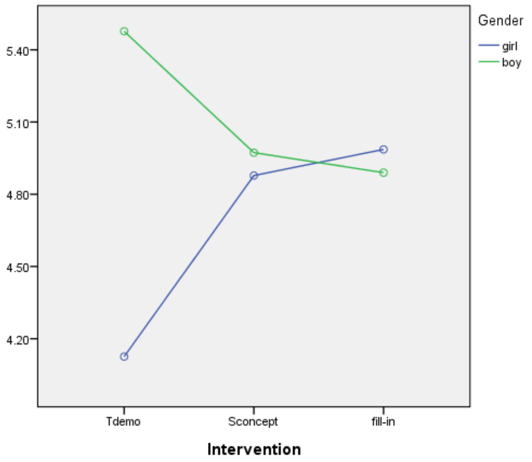


Figure 3. The Intervention-gender Interaction Plot with the Dependent Variable of Meaningfulness

4.3 Conclusion

This study concluded that gender interacting with the intervention significantly influenced learners' perceived meaningfulness of programming. However, such an effect on learners' programming performance was not supported. The study limited to only two-week implementation. Although the participants were engaged in six runs of problem-solving, elementary school students still need to be persistently engaged in the problem-solving process. Future research is suggested to extend the length of experiment and explore the impact of scaffolds on learners' change of perceived problem-solving and performance in solving computational problems. Furthermore, as PBL emphasizes collaborative learning, future research is suggested to explore whether the composition of teams or learners' attitudes toward collaboration may moderate the above-mentioned effects.

Acknowledgements

This paper was funded by research grant from the National Science Council, Taiwan, ROC (MOST 107-2511 -H-032-003-). The author would like to thank research assistants, Jia-Sin Chen, Szu-Ying Lin, Yi-Hsuan Wang for their assistance during the learning system design and data collection process.

References

Bayman, P., & Mayer, R. E. (1988). Using conceptual models to teach BASIC computer programming. Journal of Educational Psychology, 80, 3, 291–298.

Corbalan, G., Kester, L., & Van Merriënboer, J. J. G. (2009). Combining shared control with variability over surface features: Effects on transfer test performance and task involvement. *Computers in Human Behavior, 25*, 290–298.

Duncan, C., & Bell, T. (2015, November). A pilot computer science and programming course for primary school students. In Proceedings of *the workshop in primary and secondary computing education* (pp. 39-48). ACM

Heppner, P. P., & Petersen, C. H. (1982). The development and implications of a personal problem-solving inventory. *Journal of counseling psychology, 29(1)*, 66.

Hmelo, C. E., & Evensen, D. H. (2000). Introduction. Problem-based learning: Gaining insights on learning interactions through multiple methods of inquiry. In D. H. Evensen & C. E. Hmelo (Eds.), *Problem-based learning: A research perspective on learning interactions*. Mahwah: Lawrence Erlbaum Associates Publishers.

Kong, S. C., Chiu, M. M., & Lai, M. (2018). A study of primary school students' interest, collaboration attitude, and programming empowerment in computational thinking education. *Computers & Education, 127,* 178-189.

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science, 12* (2), 257-285.

Van Merriënboer, J. J. G. (2007). Alternate models of instructional design: Holistic design approaches and complex learning. In R. A. Reiser, & J. V. Dempsey (Eds.), *Trends and issues in instructional design and technology* (pp. 72–81). Upper Saddle River, NJ: Pearson/Merrill Prentice Hall.

Van Merriënboer, J. J. G., Kirschner, P. A., & Kester, L. (2003). Taking the load of a learners' mind: Instructional design for complex learning. *Educational Psychologist, 38*, 5–13.

Van Merriënboer, J. J. G., & Kester, L. (2007). Whole-task models in education. In J. M. Spector, M. D. Merrill, J. J. G. van Merriënboer, & M. P. Driscoll (Eds.), *Handbook of research on educational communications and technology*, (pp. 441–456, 3rd ed.). Mahwah, NJ: Erlbaum/Routledge.

WORKSHOP 13 - Design for Choreographies/Ambiance for Global Agile Learning to Foster Future Skills