Yang, J. C. et al. (Eds.) (2018). Proceedings of the 26th International Conference on Computers in Education. Philippines: Asia-Pacific Society for Computers in Education

Teaching Programming Languages using Robots based on Experiential Learning

Kuo-Chen LI*, Shih-Ming PI & Hsiang-Ting LIN
Department of Information Management, Chung Yuan Christian University, Taiwan
*kuochen@cycu.edu.tw

Abstract: In STEM education, computational thinking is a possible way to help solve real-life problems. Learning computer programming is an effective way to develop computational thinking skills. In this paper, we propose a learning model for programming languages based on experiential learning theory. The use of robots carries out the hands-on experiences during the learning process. The empirical experiment shows that the experiential learning model proposed in this study can effectively affect children's interest in learning.

Keywords: STEM, computational thinking, experiential learning, Scratch, robotic application

1. Introduction

According to ISTE (2015), computational thinking is a possible way to help solve real-life problems. Using human creativity and critical thinking, computer programmers can improve the ability to solve critical human problems. To succeed in computer programming, programmers must first develop and acquire advanced thinking skills such as problem solving, logic and mathematical thinking, critical thinking and creative thinking. On the other hand, learning computer programming is also an effective way to develop these skills. In STEM (Science, Technology, Engineering, and mathematics) education, computational thinking also plays an important role. In addition, hands-on practice from students' STEM learning experience is addressed as important links to the acquired knowledge (Connor, Ferri, & Meehan, 2013).

In recent years, programming education at primary school has prospered, and many visual programming tools have been proposed for primary school students, such as LEGO Robotics, Scratch, Blockly, Kodu, Alice, etc. These visual programming languages are mostly easy to understand and graphically designed. According to Çayır (2010), using LEGO for computational thinking education has been successfully in introducing students to real life experiences, providing cooperative learning opportunities for students and allowing them to use their interdisciplinary knowledge to help solve the problem. The robotic application makes it easier for students to learn the concepts of science, technology and engineering (Moore, 1999; Papert, 1980).

Based on the previous studies, those visual programming languages have a significant improvement in the motivation. However, different teaching models might vary the learning outcome on the abilities such as logical reasoning ability, problem solving ability, and creative thinking. In this study, we tend to propose an experiential learning model for Scratch using Lego robots and evaluate the effectiveness of the proposed model.

2. Related Works

2.1 Computational Thinking

Computational thinking is considered the heart of all STEM disciplines (Henderson, Cortina, Hazzan, & Wing, 2007) and is the key process involved in developing problems and solutions (Wing, 2011). To train the computational thinking, learning programming languages is always an effective way. The programming process is a process of solving problems. It is closely related to high-level thinking skills such as problem solving, logic and mathematical thinking, critical thinking and creative thinking (Korkmaz & Altun, 2014). Programming is not only a basic skill in computational science, but also a key tool in supporting computational thinking (Grover & Pea, 2013).

As the modern economy is deeply influenced by technologies, acquiring computatinal thinking is critical to the success of the next generation of students. Computational thinking is increasingly seen as an important component of STEM learning in primary and secondary education (Eguchi, 2014). Since most primary and secondary school students nowadays can easily use a variety of electronic products, teachers should use this advantage to teach students computational thinking using these devices (Wing, 2008). Many scholars have suggested that computational thinking should include abstraction, problem decomposition, modeling and simulation, and algorithmic thinking (Barr & Stephenson, 2011; Grover & Pea, 2013; Selby & Woollard, 2014; Wing, 2011).

2.2 Experiential Learning Theory

Experiential Learning Theory is defined as "the process of creating knowledge through the transformation of experience, and the acquisition of knowledge from the combination of understanding and transforming experiences" (Kolb, 1984). The learning processes require the active participation of students, which is often associated with teacher teaching and the class curricula (Clark, Threeton, & Ewing, 2010).

Kolb's experiential learning theory provides a clear instructional design mechanism that emphasizes constructivist views on how people construct knowledge. Kolb believes that in order to have a complete learning experience, students must go through a four-stage learning cycle that not only allows students to fully investigate a topic through different activities and perspectives, but also adapts to different learning styles. According to Kolb, these learning styles are the product of doing and observing, thinking and feeling. The processes of experiential learning are divided into: 1. Concrete Experience; 2. Reflective Observation; 3. Abstract Conceptualization; 4. Active Experimentation as shown in Fig.1.

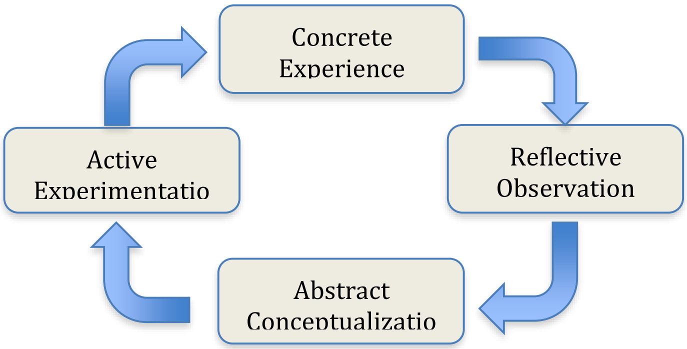

Figure 1. Kolb's experiential learning cycle

The methods and techniques of experiential learning theory in educational contexts include group discussions, group responses, listening reports, demonstrations, case studies, games, debates, etc (Lee & Caffarella, 1994). It can be seen from the above that experiential learning is not limited to outdoor activities. It can be used to learn indoors and use diverse methods and techniques to create a new learning style for learners to facilitate learning and integration.

3. Proposed Model

3.1 Model design

The proposed model is designed with Kolb's experiential learning theory. Kolb's experiential learning theory provides a clear instructional design mechanism that emphasizes how people construct knowledge and allows children to learn through experiences. Fig. 2 shows the proposed model based on experiential learning theory. At the concrete experience step, students will construct and play with the robots to gain the physical experiences. After the robot is built, students can use the demo program to see how the robot is controlled. In this phase, students can build up concrete experience through the hands-on practice. After gaining the experiences, students will observe and learn the corresponding logic and syntax at reflective observation step. According to the building blocks instructions and concepts learned in this class, students now are able to enter the next phase. At conceptualization phase, students will try the building block instructions and let the robot perform small tasks. Based on previous specific experiences and observational reflections, students will construct corresponding concepts that can be used to solve the problem. The final step, active experimentation, students will use what they have learnt from the previous steps to re-produce the task they experienced at the first step.

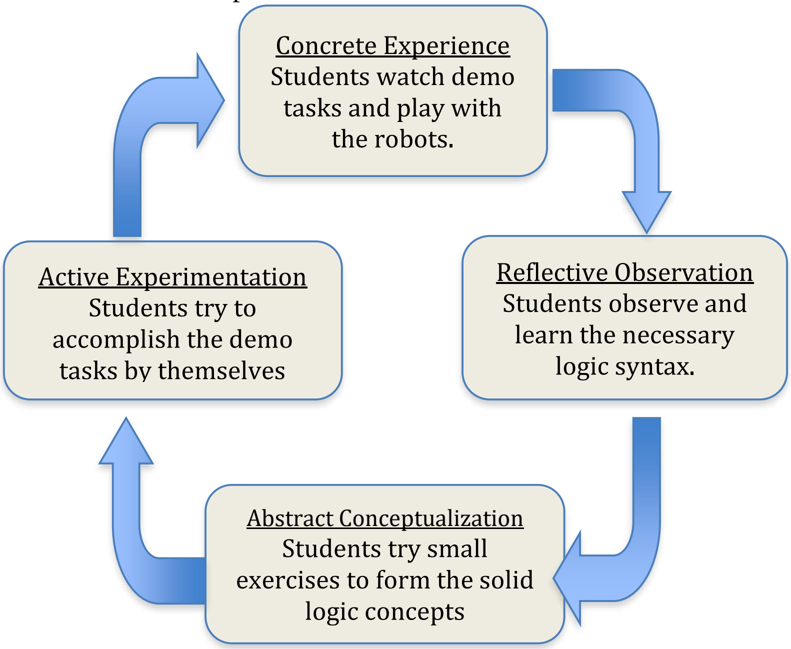

Figure 2. Proposed model based on experiential learning

Many studies have shown that robotic applications can enhance students' problem-solving skills, computational thinking skills (Varney, Janoudi, Aslam, & Graham, 2012; Zaharija, Mladenović, & Boljat, 2013). The proposed model uses robotics application as the media to carry out the experiential learning to enhance the computational thinking performance of the graphical programming language teaching. In addition, using robots in the classroom offers a unique opportunity to change the classroom atmosphere (Piteira, 2011) and gives learners the opportunity to "solve" real-world problems and enhance abstract programming concepts.

3.2 Course design

Based on the proposed model, a course plan, which teaches conditional selection logic can be designed as shown in table 1. The course design follows the four steps of experiential learning theory. Students can establish the solid experiences from playing with the robots and the demo programs. Using reflective thinking recognizes the corresponding learning content. After a series of small practices, students can form the abstract concepts. Finally, students can use the concepts to solve the problems. Fig. 3(a) shows the selection logic with the real life objects such as motors, sensor, break, etc. On the other hand, Fig. 3(b) demos the regular Scratch selection syntax block.

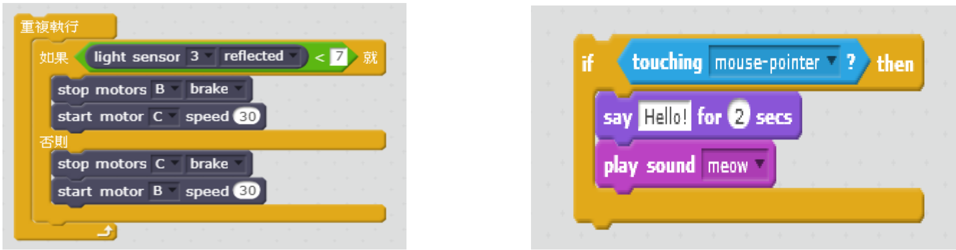

Figure 3. (a) Robot selection syntax block (b) Scratch selection syntax block

Table 1

Experiential learning	Teaching content	Time
Concrete Experience	Construct the robots. Play with the robot with demo program, which be able to turn when the robot encounters an obstacle.	10mins
Reflective Observation	Students observe the robot to figure out the selection logic. Teacher teaches the selection logic syntax block (if-else).	15mins
Abstract Conceptualization	Students try to use if-else syntax block to control the robot to do simple tasks.	10mins
Active Experimentation	Students try to duplicate the demo program using what they have learnt.	15mins

3.3 *Empirical experiment*

An empirical experiment is carried out to validate the effectiveness of the proposed framework. 20 third grader students were given the class designed from the previous section. Students have no experiences for Scratch. Pre-test and post-test were given during the class section. Fig. 4 shows the empirical experiment for the proposed model. Paired t-test is applied after the pre-test and post-test. Result shows the significance of the proposed model in Table2.

Figure 4. Empirical experiment

Table 2. Paired t-test for the empirical experiment

	mean	Std. dev	Std. error	t	df	Sig (two-tailed)
Pre-Post	-17.50	23.717	7.500	-2.333	9	.045

4. Conclusion

The empirical experiment shows that the experiential learning model proposed in this study can effectively affect children's interest in learning. In the future, other subjects of computational thinking could be applied with the experiential learning model as well. During the experiment, we found that most of the students were able to use their creativity in assembling robots. However, some of the students will focus on assembling robots more, and use less time on programming thinking. It shows that using robots sometimes might become distractions for programming learning. It is necessary to intervene in a timely manner to help students learn better to complete the program tasks.

References

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community? Acm Inroads, 2(1), 48-54.

Çayır, E. (2010). Assignment the effect of learning environment supported by lego-logo on science process skill and self concept, Master dissertation, Sakarya University, Sakarya.

Clark, R. W., Threeton, M. D., & Ewing, J. C. (2010). The potential of experiential learning models and practices in career and technical education & career and technical teacher education. Journal of Career and Technical Education, 25(2).

Connor, K. A., Ferri, B. H., & Meehan, K. (2013). Models of mobile hands-on STEM education. Department for Education in England (DOEE) (2013, September 11). National curriculum in England: Computing programmes of study.

Eguchi, A. (2014). Robotics as a learning tool for educational transformation. Paper presented at the Proceeding of 4th International Workshop Teaching Robotics, Teaching with Robotics & 5th International Conference Robotics in Education Padova (Italy).

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. Educational Researcher, 42(1), 38-43.

Henderson, P. B., Cortina, T. J., Hazzan, O., &Wing, J. M. (2007) Computational thinking. In Proceedings of the 38th ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE '07), 195–196. New York, NY: ACM Press.

ISTE (2015).It's time to demystify computational thinking. Retrieved from https://www.iste.org/explore/articleDetail?articleid=501

Kolb, D.A. (1984). Experiential learning: Experience as the source of learning and development. Englewood Cliff, NJ: Prentice-Hall.

Korkmaz, Ö., & Altun, H. (2014). Adapting Computer Programming Self-Efficacy Scale and Engineering Students' Self-Efficacy Perceptions. Online Submission, 1(1), 20-31.

Lee, P., & Caffarella, R.S. (1994). Methods and techniques for engaging learners in experiential learning activities. New Directions for Adult and Continuing Education, 62, 43-54.

Moore, V. S. (1999). Robotics: Design through geometry. The Technology Teacher, 59(3), 17–22.

Papert, S. (1980). Mindstorms: Children, computers, and powerful ideas. New york: Basic Books, Inc.

Piteira, M., & Haddad, S. R. (2011). Innovate in your program computer class: an approach based on a serious game. Paper presented at the Proceedings of the 2011 Workshop on Open Source and Design of Communication.

Selby, C., & Woollard, J. (2014). Refining an understanding of computational thinking.

Varney, M. W., Janoudi, A., Aslam, D. M., & Graham, D. (2012). Building young engineers: TASEM for third graders in Woodcreek Magnet Elementary School. IEEE Transactions on Education, 55(1), 78-82.

Wing, J. M. (2008). Computational thinking and thinking about computing. Philosophical transactions of the royal society of London A: mathematical, physical and engineering sciences, 366(1881), 3717-3725.

Wing, J. (2011). Research notebook: Computational thinking—What and why? The Link Magazine, Spring. In: Carnegie Mellon University, Pittsburgh.

Zaharija, G., Mladenović, S., & Boljat, I. (2013). Introducing basic programming concepts to elementary school children. Procedia-social and behavioral sciences, 106, 1576-1584.