

The Effects of AR-based Instruction on Students' Learning Performance, Motivation and Self-efficacy in Programming Learning

Gloria Yi-Ming KAO* & Cheng-An RUAN

Graduate Institute of Digital Learning and Education, National Taiwan University of Science and Technology, Taiwan

*gloriakao@gmail.com

Abstract: Recent studies have reported benefits of Augmented Reality (AR)-based instruction in various learning domain. However, few studies were done to explore its effects in programming learning. In this study, we devolved an AR-based instruction with high (puzzle cards) or low (fixed card) interaction levels to assist programming learning. The results showed that students' learning performance and self-efficacy were improved after the experiment, which indicated AR-based learning did have positive effects though no difference between high and low interaction could be determined at this stage.

Keywords: augmented reality, programming learning, learning performance, motivation, self-efficacy

1. Introduction

1.1 Augmented Reality (AR)

Augmented Reality (AR) is a technique that adds virtual objects in a real environment, which can redeem the lack of information in a real environment (El Sayed, Zayed, & Sharawy, 2010). Physical and virtual tools are commonly utilized as assistant teaching tools in classrooms; however, it is hard to combine the strengths of the two tools. After AR was invented, which integrated physical and virtual functions, learners were supplied with better learning experiences (Bujak et al., 2013). AR enables learners get close to real environments from learning environments and supplies richer sensory experiences. Also, it makes learners have the opportunities to operate physical objects and then to interact with virtual ones (Wojciechowski & Cellary, 2013).

1.2 Programming Learning

Programming is a big role in science, technology, engineering and mathematics (STEM) fields. Especially, to those major in computer science, programming course plays an important role. Nevertheless, learning programming is difficult to the beginners who are not in the related fields (McCracken et al., 2001). Despite of learners' ages, programming is always difficult for beginners (Kelleher & Pausch, 2005). In Taiwan, traditional way to learn programming usually use textbooks or run sample program codes with computers, and then observe the execution results. However, beginners are not able to realize the programming process or the results. This can lead to poor learning motivation or performance.

1.3 AR in Education

Recently, many studies have shown that AR-based learning has positive influence on students' learning, such as learning performance (Lin, Duh, Li, Wang, & Tsai, 2013), learning motivation (Di Serio, Ibáñez, & Kloos, 2013) and self-efficacy (Kamarainen et al., 2013).

Therefore, AR is commonly used as an educational tool. Liu and Tsai (2013) employed teaching materials combined with AR in an English writing course, trying to reduce the difficulties of the students while learning second language. Results indicated that learners under this situation could construct contents and knowledge much easier; moreover, more meaningful articles were produced, and language learning performance was then increased. Chang, Chang, Sung, Chao and Lee (2014) developed an AR guide system and then used it in an art appreciation course. Compared to general audio guide and non-guide environment, learners in AR guide system group had more fluent experience and better learning performance. Chang, Wu and Hsu (2013) stimulated Fukushima nuclear disaster and explored the situation of nuclear pollution by using AR technique. The result show that AR based environment could improve students' comprehension and increase their sensorial immersion. Ibáñez, Di Serio, Villarán and Delgado Kloos (2014) implemented AR in a basic course of electromagnetism. The results showed that, in this environment, students could not only understand the phenomena and concepts of electromagnetism more efficiently but reach higher flow experience levels, compared to web-based learning environment. AR is used in many different subjects, yet cases of programming course are seldom found. Consequently, this study explores how AR-based learning influence students' learning performance and motivation in a programming course.

2. System design

2.1 Software Development

Aurasma, a cross-platform AR development system developed by Hewlett-Packard Development Company in 2011, supplies simple operating interface to make it easier for developers to produce AR contents. Aurasma enables users to connect to the database to get the latest AR contents by using the mobile device.

Based on the Aurasma, we devolved an AR-based application with high (puzzle cards) or low (fixed card) interaction levels to assist programming learning. Students could freely assemble puzzle cards to observe the corresponding results. With high interaction level, students are encouraged to try and figure out what combinations could work or not. With low interaction level, students could only watch default animations with fixed cards. We used pictures from the Scratch program, developed by Massachusetts Institute of Technology in 2003, in our design.

2.2 Operating process

Learners followed the teacher instruction, assembled different puzzle cards (as Figure 1 shows), and used the Aurasma in mobile device to observe and compare the differences among various combinations of puzzle cards. While the puzzle cards are assembled correctly, the programming operating animation will be displayed (as Figure 2 shows). However, the animation will not be displayed when the assembling is not correct.

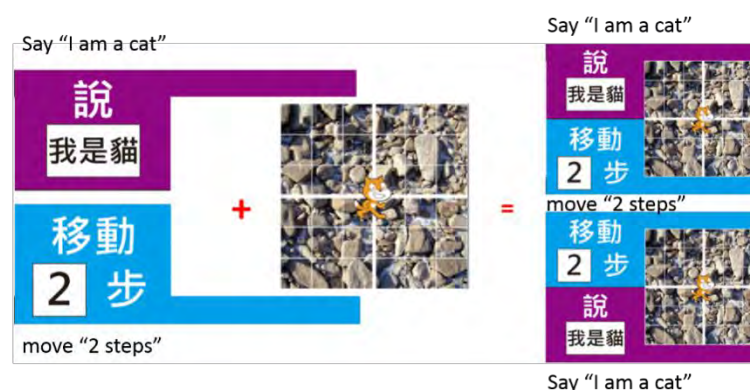


Figure 1. Assemble different puzzles.



Figure 2. The animation is displayed when the assembling is correct.

2.3 Learning Content

The learning content of the study was three main structures of flowchart (as Figure 3 shows) in computer programming. They were 1) Sequence structure, which followed certain order, operating the description separately; 2) Selection structure, which operates the program according to the conditional determination; 3) Iteration structure, which operates the descriptions repeatedly, until the descriptions match the breaking condition, the operation ends. Most Algorithms can be consisted of these three structures. This course is not only an important role but an essential part in programming courses. Several combinations of puzzles were supplied. Puzzles can be assembled with two combinations in Sequence and selection structures, but they can be assembled with three combinations in Iteration structure.

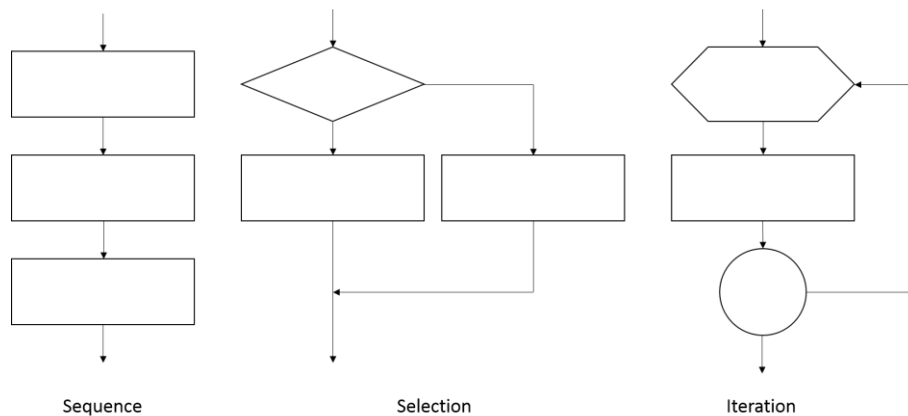


Figure 3. Three main structures of flowchart.

3. Experiment design

3.1 Participants

Sixty-five seventh graders from two mixed-ability classes in a junior high school in North Taiwan participated in the experiment, the mean age was 13. Thirty-one (15 females and 16 males) were in the experimental group, while thirty-four students (16 females and 18 males) were in the control group. High interactive AR-based learning systems (puzzle cards) were used in the experimental group, while low interactive AR-based learning system (fixed cards) was in the control group.

Mobile devices were not used in previous teaching process, so the students were instructed to operate the mobile devices and the AR system first before the experiment was carried out. In both of the two groups, there were 7 students had learned Scratch, a software which can program interactive

animations. Their average learning time was one year. Therefore, a pretest was adopted to exclude the differences among the students caused by prior knowledge.

3.2 Learning activities and environment

In order to fit in the teacher's original teaching style, the experiments were taken place in a computer classroom. During the experiment, computers were only used for broadcasting PowerPoint slides for instructional purpose. For both the experimental and control groups, each two students were equipped with one mobile device (iPad) and were allowed to discuss with each other.

As Figure 4 shows, high interactive AR-based learning systems were used in the experimental group. This group was given puzzle cards and was asked to freely assemble those puzzles to yield and observe different program execution outcomes. While students in the control group with low interactive AR-based learning system could only get fixed cards and observe the default program execution results.

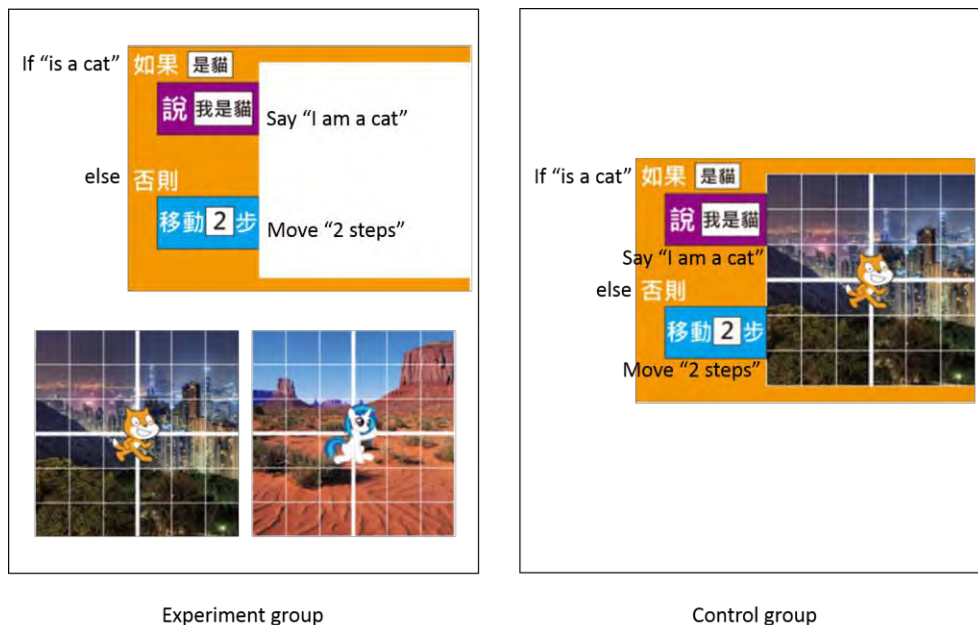


Figure 4. Experiment group with puzzle cards and control group with fixed cards.

3.3 Experiment procedure

Figure 5 shows the four stages of this study and describe as below:

- Stage 1: A pretest was adopted to examine the students' prior knowledge about programming; pre-questionnaires were employed to probe students' learning motivation and self-efficacy. Students were firstly explained the ways to fill out the sheets and were not allowed to discuss during the test. This stage took ten minutes.
- Stage 2: With one mobile device equipped, two students grouped and were taught how to operate the mobile devices and the learning system. This stage took ten minutes. The researcher checked the students' status to make sure they were able to operate the tools.
- Stage 3: Programming course was started. There were four main units in this course. Students observed and learned with mobile devices after the teacher's instruction. This stage took sixty minutes.
- Stage 4: A learning performance posttest, a learning motivation post-questionnaire and self-efficacy post-questionnaire were adopted. Students were not allowed to discuss during the test. This stage took twenty minutes.

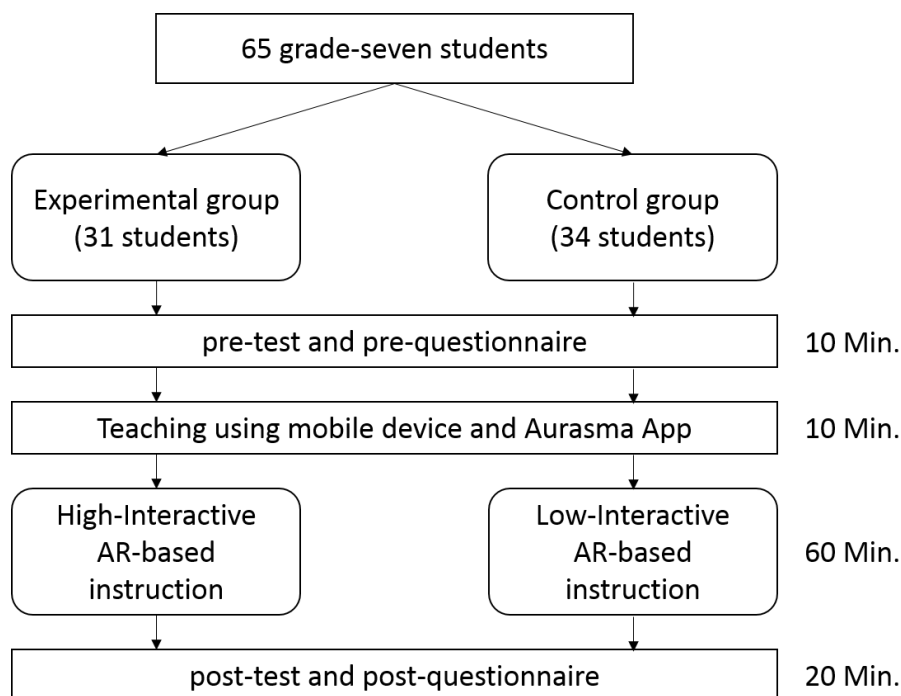


Figure 5. Experiment procedure.

4. Analysis methods and results

4.1 Analysis tools

Learning performance test: A self-edited performance test sheet was used, and the expert validity was constructed after the scale was revised by three computer and information science teachers. A pretest was employed to examine the students' prior knowledge about programming. Each of the ten matching questions in the sheet was ten points. A post-test was adopted to explore students' knowledge and applied abilities about programming. There were five matching questions and five applicant questions, and each question was ten points. From this test, we wanted to understand if the learning performance was improved after the experiment.

Learning motivation and self-efficacy: The learning motivation questionnaire adopted in the study was a revision from Hwang, Yang and Wang (2013) and included seven questions. The self-efficacy questionnaire was a revision from Wang and Hwang (2012) which included eight questions. Both the questionnaire adopts five-point Likert rating scheme, "1" means very disagree while "5" means very agree. All the descriptions in the questionnaire were positive (e.g. I think this course is meaningful and worthy learning, I believe that I can get great score in the assignments). For the purpose to explore the difference before and after the intervention, the two questionnaires were employed both before and after the experiment to investigate students' perceptions toward learning motivation and self-efficacy. The original Cronbach's alpha of the motivation questionnaire was 0.79, while the revised one was 0.883. The original Cronbach's alpha of the self-efficacy questionnaire was 0.916, while the revised one was 0.919.

4.2 Method

The present study analyzed and processed the data by using statistical software. Firstly, the descriptive statistic, mean and standard deviation, would show the differences among learning performance, learning motivation and self-efficacy. Secondly, after pre-test and pre-questionnaire, independent t-Test will be employed to examine if there are any differences between the two groups. Thirdly, ANCOVA will be used to explore if there are any differences in post-test and post-questionnaire between the two groups. Above three steps will be adopted to observe how high and low interactions influence AR-based instruction. Lastly, how AR-based instruction influence programming learning will be discussed. Also,

dependent t-Test will be used to examine whether there are any significant differences between pre-and-post test, and pre-and-post questionnaire.

4.3 Results

Several independent t-Tests were conducted to examine the difference between experimental and control group before the experiment. As shown in Table 1, there is no significant difference between experimental and control group (pre-test: $t=-0.403$, $p>0.05$, pre-motivation: $t=1.103$, $p>0.05$, pre-self-efficacy: $t=0.136$, $p>0.05$) which means before the learning activity, students in the two groups have equivalent prior knowledge, learning motivation and self-efficacy.

Table 1: Independent t-Test result of Pre-test, Pre-motivation and Pre-self-efficacy of Experimental group and Control group.

		N	Mean	SD	t	Sig.
Pre-test	Experimental group	31	47.097	19.008	-0.403	0.689
	Control group	34	48.824	15.524		
Pre-motivation	Experimental group	31	4.143	0.655	1.103	0.274
	Control group	34	3.975	0.574		
Pre-self-efficacy	Experimental group	31	3.347	0.948	0.136	0.892
	Control group	34	3.320	0.582		

After the learning activity, several analysis of covariance (ANCOVA) were used to evaluate the difference between experimental and control group in terms of learning performance, motivation or self-efficacy. The pre-test score of each evaluation was used as covariate while the post-test score of each evaluation was used as dependent variable. Table 2 shows that the results are not significant (post-test: $F=0.082$, $p>0.05$, post-motivation: $F=1.216$, $p>0.05$, post-self-efficacy: $F=0.142$, $p>0.05$) which means high or low interactive AR-based learning would not impact on students' learning performance, motivation or self-efficacy.

Table 2: ANCOVA result of Post-test, Post-motivation and Post-self-efficacy of Experimental group and Control group.

		N	Mean	SD	F	Sig.
Post-test	Experimental group	31	69.839	19.701	0.082	0.775
	Control group	34	68.618	17.132		
Post-motivation	Experimental group	31	4.014	0.654	1.216	0.274
	Control group	34	3.870	0.554		
Post-self-efficacy	Experimental group	31	3.581	0.935	0.142	0.707
	Control group	34	3.522	0.607		

Several dependent t-Tests were conducted on exploring how AR-based learning influences students' learning performance, motivation and self-efficacy, as shown in Table 3. There is a significant difference between learning performance ($t=-8.634$, $p<0.05$) and self-efficacy ($t=-2.557$, $p<0.05$), yet learning motivation is not significantly different ($t=1.559$, $p>0.05$). The results showed overall after the learning activity, students' learning performance and self-efficacy were increased significantly, although no difference between the puzzle cards and fixed cards could be determined.

Table 3: Dependent t-Test result of Pre-test, Post-test, Pre-motivation, Post-motivation, Pre-self-efficacy and Post-self-efficacy.

	N	Mean	SD	t
Pre-test	65	48.000	17.157	-8.634***
Post-test	65	69.200	18.266	
Pre-motivation	65	4.055	0.615	1.559
Post-motivation	65	3.938	0.603	

Pre-self-efficacy	65	3.332	0.772	-2.557*
Post-self-efficacy	65	3.550	0.775	

*** $p < .001$, * $p < .05$

5. Conclusions and discussion

The present study explores how high and low interactive AR-based learning influence students' learning. As the results showed, different interactive levels with puzzle or fixed cards did not impact on students' learning performance, motivation and self-efficacy. Instead, overall students' learning performance and self-efficacy were improved after the experiment, which indicated the innovative use of AR into programming learning did have positive effects although no difference between high and low interaction could be determined at this stage.

Convenience sampling was conducted in this study, which was a limitation of the study. The results can only represent the learning performance of the students in the experiment, but cannot infer the overall students' in other areas. Further, time duration was another limitation of the study. Owing to the time, the states of students' learning and how much did they exactly learned were not thoroughly considered.

Future studies are suggested dividing the participant groups into high-interactive AR, low-interactive AR, and traditional learning to further investigate how AR-based learning influences students' learning.

Acknowledgements

This research was funded by the National Science Council of the Republic of China, Taiwan, under contract number NSC 102-2511-S-011 -005 -MY3.

References

- Bujak, K. R., Radu, I., Catrambone, R., MacIntyre, B., Zheng, R., & Golubski, G. (2013). A psychological perspective on augmented reality in the mathematics classroom. *Computers & Education*, 68, 536-544.
- Chang, K. E., Chang, C. T., Hou, H. T., Sung, Y. T., Chao, H. L., & Lee, C. M. (2014). Development and behavioral pattern analysis of a mobile guide system with augmented reality for painting appreciation instruction in an art museum. *Computers & Education*, 71, 185-197.
- Chang, H. Y., Wu, H. K., & Hsu, Y. S. (2013). Integrating a mobile augmented reality activity to contextualize student learning of a socioscientific issue. *British Journal of Educational Technology*, 44(3), E95-E99.
- Di Serio, Á., Ibáñez, M. B., & Kloos, C. D. (2013). Impact of an augmented reality system on students' motivation for a visual art course. *Computers & Education*, 68, 586-596.
- El Sayed, N. A., Zayed, H. H., & Sharawy, M. I. (2011). ARSC: Augmented reality student card. *Computers & Education*, 56(4), 1045-1061.
- Hwang, G. J., Yang, L. H., & Wang, S. Y. (2013). A concept map-embedded educational computer game for improving students' learning performance in natural science courses. *Computers & Education*, 69, 121-130.
- Ibáñez, M. B., Di Serio, Á., Villarán, D., & Delgado Kloos, C. (2014). Experimenting with electromagnetism using augmented reality: Impact on flow student experience and educational effectiveness. *Computers & Education*, 71, 1-13.
- Kamarainen, A. M., Metcalf, S., Grotzer, T., Browne, A., Mazzuca, D., Tutwiler, M. S., & Dede, C. (2013). EcoMOBILE: Integrating augmented reality and probeware with environmental education field trips. *Computers & Education*, 68, 545-556.
- Kelleher, C., & Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys (CSUR)*, 37(2), 83-137.
- Lin, T. J., Duh, H. B. L., Li, N., Wang, H. Y., & Tsai, C. C. (2013). An investigation of learners' collaborative knowledge construction performances and behavior patterns in an augmented reality simulation system. *Computers & Education*, 68, 314-321.
- Liu, P. H. E., & Tsai, M. K. (2013). Using augmented- reality- based mobile learning material in EFL English composition: An exploratory case study. *British Journal of Educational Technology*, 44(1), E1-E4.

- McCracken, M., Almstrum, V., Diaz, D., Guzdial, M., Hagan, D., Kolikant, Y. B. D., ... & Wilusz, T. (2001). A multi-national, multi-institutional study of assessment of programming skills of first-year CS students. *ACM SIGCSE Bulletin*, 33(4), 125-180.
- Wang, S. L., & Hwang, G. J. (2012). The role of collective efficacy, cognitive quality, and task cohesion in computer-supported collaborative learning (CSCL). *Computers & Education*, 58(2), 679-687.
- Wojciechowski, R., & Cellary, W. (2013). Evaluation of learners' attitude toward learning in ARIES augmented reality environments. *Computers & Education*, 68, 570-585.