Volume 13, Number 1, 2011

Global Journal of Engineering Education

An alternative learning strategy to support engineering students' programming skills: a case study

Pao-Nan Chou† & Hsi-Chi Hsiao‡

National University of Tainan, Tainan, Taiwan†
Cheng Shiu University, Kaoshiung, Taiwan‡

ABSTRACT: The purpose of this study was to evaluate the instructional effectiveness of a game-based learning strategy, and to construct a role-play learning framework for engineering students. Forty-two undergraduate students, majoring in computer sciences and information engineering participated in this study. A mixed methods research design was employed to collect data. Through the results of an experimental analysis, students who studied in the game-based approach performed better than those who received a non game-based activity ($t=3.49$, $p<0.01$). Interviews with members of the game-based group resulted in the creation of a six-stage learning framework, which illustrates how students cognitively engaged in the role-play activity.

Keywords: Role-play strategy, mixed methods study, game-based learning, alternative engineering instruction

INTRODUCTION

Learning a programming language (e.g. C++) is often regarded as a difficult task for school students in different disciplines [1]. From a psychological perspective, students must engage in a complex cognitive learning process during program design and development [2]. For example, program debugging requires several problem-solving skills and program writing involves varied organisation strategies. Therefore, how to impart programming knowledge to students effectively, especially for programming pedagogy, becomes a thorny issue for course instructors [3].

An abundance of previous studies reported that Game-Based Learning (GBL) in classrooms might enhance students' learning performances within various educational levels of skills [4]. Compared to a traditional lecture-based learning mode, under the context of GBL, students learn more efficiently not only from factual information, but also by reasoning-related knowledge [5]. Recently, Mayo contended that GBL has the potential to improve overall learning outcomes of engineering students [6]. For the previously mentioned reason, whether or not integrating GBL into a programming curriculum would benefit engineering students' cognitive learning is worthy of exploration.

However, a review of existing literature regarding the adoption of GBL in course teaching reveals potential problems; a summary of which is as follows:

1. Non cost-efficient adoption: Most past studies tended to employ game-based simulators or educational gaming software to support students' learning (e.g. Ref. [7]). Although significant learning outcomes resulted in this type of game-designed research, a potential problem is that development and implementation of games for specific levels of school students may consume significant amount of educational resources. For teaching and learning purposes, integrating gaming into curricula may not be a cost-efficient approach for course instructors.
2. By-product problems: Some past studies attempted to use available commercial games in the market to evaluate students' learning outcomes (e.g. Ref [8]). Even though these exciting games motivated students and enhanced students' learning spans, a potential problem is that in the long run, students may become addicted to the features of the games rather than instructional contents. Furthermore, for school educators, in order to prepare courses successfully, extra expenditure on advanced hardware, such as upgrading a graphics-accelerator card, are necessary.

To overcome these problems, the current study's design is a learning strategy to allow students to situate themselves in a game-based environment. The strategy has its basis in educational gaming's role-play features, which aim to strengthen students' cognitive development for a selected curriculum [1][4][6]. One mixed-method research model, consisting of quantitative and qualitative methods evaluates the targeted learning strategy.

From the aspect of the quantitative approach, one randomised-based experimental study with post-test design explores engineering students' learning performances by implementing a role-play activity in a programming course. In contrast to the qualitative method, a phenomenological study elicits engineering students' perceptions of learning processes in the role-play activity. Specifically, the purpose of the current research is to evaluate the instructional effectiveness of a game-based learning strategy, and to construct a role-play learning framework for engineering students.

RESEARCH METHODS

Case Study Description

1. Course Content: Advanced Dynamic Programming (3 credits) is a course offering of the Department of Computer Sciences and Information Engineering at a southern Taiwanese university. This course focuses on advanced knowledge of Web-based dynamic programming and basic integration of on-line databases. The programming language and database system used in the course are PHP and MySQL, respectively.
2. Course Instructor's Experience: The course instructor has offered several programming-related courses at the university for several years. Despite practical teaching experience, the instructor's instructional style in programming classes follows a traditional method: lecture in the classroom and practice in the computer laboratory. Recently, due to trends from professional instructional development, the instructor consults the research for innovative learning strategies.
3. Participants: Forty-two college students (Class A: 20 and Class B: 22), majoring in computer science and information engineering, participated in this study. Students' average age was 21 years, and they had completed an introductory course of basic Web-based programming. In this study, Class A engaged in the role-play activity.
4. Role-Play Scenario: Each team has two members. The partners are experienced programming designers at a software company. Currently, in order to decrease workload, the company needs to hire new programmers who excel at Web-based programming, especially involving PHP and MySQL design. Players' responsibility is staff recruitment. Partners' job for selecting qualified applicants is to develop ten problem-solving programming questions, which emphasise code debugging.
5. Programming Questions: The content of programming questions must cover what students learned during the previous eight weeks of class. Each team must ensure the quality of programming questions, which lack syntax errors exist. In order to assist students' design process, the teaching Web site contained a design manual, which explains how to design programming questions.
6. Implementation of Role-Play Activity: A specific day was set for activity implementation. A week prior to activity day, each team submitted ten questions to the course instructor. On activity day, the role of each team shifted to job applicants. Each team randomly received ten questions prepared by other teams. At this stage, the goal of team members in each team was to solve the ten coding questions collaboratively within three hours. After the activity, each team provided correct answers for the questions they designed. Table 1 summarises the details of the role-play activity.

Table 1: Details of the role-play activity.

Sequence	Description
1 (8-week class: normal instruction)	Students in Class A received normal instruction over eight weeks. In the last week, students received notification of an instructional activity to ensue in the succeeding week. The course instructor illustrated role-play activity details.
2 (1-week preparation: no class)	Students chose their team members and began to design the programming questions during one week. Subsequently, each team submitted designed questions to the course instructor.
3 (Activity in the class)	On activity day, each team completed a practice test prepared by another team. Team members collaboratively solved the questions in the test.

Quantitative-Based Design

1. Treatment Group: This study developed two treatments. Students in Class A received a role-play activity, while students in Class B engaged in normal instruction (non role-play activity), in which students only discussed among themselves what they learned during the previous eight weeks of class. One research null hypothesis became: *No statistically significant differences exist between two instructional treatments*. Figure 1 depicts the study's experimental design.

2. Independent Variables: The study examined one independent variable: Type of instructional activity (GBL-related and non GBL-related activities).
3. Dependent Variables: One week after the instructional intervention, one measurement instrument (post-test) developed by the course instructor assessed students' knowledge about PHP programming and MySQL database. The measurement consists of twenty problem-solving questions and reflects high validity and reliability since the instructor had used it for several academic years.
4. Data Analysis: The Statistical Package for the Social Sciences (SPSS) completed the study's data analysis. Two statistical techniques, Independent T-Test and Cohen's Effect Size, analysed the collected data. The techniques' purposes were to compare mean differences, and evaluate practical significances in the two instructional treatments. An alpha value (0.05) was set to test the research's hypothesis.

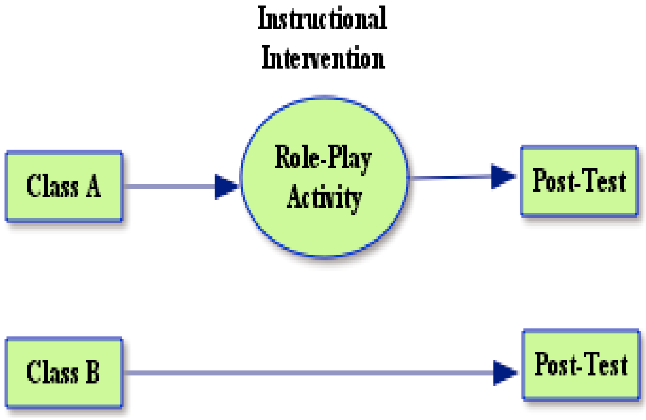

Figure 1: Experimental design of the current study.

Qualitative-Based Design

1. Evaluation Method: This study also adopts Van Manen's phenomenology methodology to explore students' lived experiences in the role-play learning activity [9]. A possible interpretation of participants' perceptions of engagement in the instructional intervention was constructed.
2. Evaluation Process: After students completed the instructional intervention, the researchers interviewed each team in Class A through an informal discussion. The interviews focused on students' learning perspectives during the role-play activity. The interview process lasted at least one hour.
3. Data Analysis: A retained collaborator transcribed the interview data. Subsequently, Moustakas's method of phenomenological analysis analysed all written transcripts [10]. The analysis techniques include:

- Preliminary grouping;
- Clustering of invariant meaning units;
- Searching for themes;
- Compositing textural-structural descriptions; a computer program called Nvivo supported the qualitative data analysis.

RESULTS AND DISCUSSION

Quantitative Findings

Table 2 reports a summary of descriptive statistics regarding the post-test.

Table 2: Summary of descriptive statistics for post-test.

Group	Number of Subjects	Mean	Standard Deviation	Minimum Value	Maximum Value
Class A	20	84.25	10.42	70	100
Class B	22	72.50	11.31	55	95

Score in the post-test ranges from 0 to 100

From the results shown in Table 2, on average, students in Class A obtained 84.25 points (SD = 10.42) in the post-test while students in Class B obtained 72.50 points (SD = 11.31) in the post-test.

In this study, significant (p) value was set to 0.05 and inferential statistics was run at a two-tailed mode. Table 3 reports the results of the T-test.

Table 3: Results of T-Test.

	t	DF	Significant	Mean Difference
Between Group	3.49	40	*0.001	11.75

*$p < 0.01$

From the results in Table 3, an effect from instructional treatment appears from the post-test ($t = 3.49$, $p < 0.01$). A significant difference appeared for the two treatment groups. By using a formula (Cohen's d= 2t/DF) with the results of the T-test, Cohen's effect size is 1.10, which indicates that a large difference exists between the experimental group (role-play) and the control group.

In summary, students in Class A significantly out-performed those in Class B. The role-play learning strategy, indeed, enhanced students' learning performances. Therefore, the research hypothesis in this study was rejected at a 0.01 significant level.

Qualitative Findings

During transcript interpretation, the research focus of the qualitative inquiry is students' learning process during the role-play activity. Information not related to this category was excluded. Overall, qualitative analysis resulted in three themes whose details, with representative quotations from participants, are summarised as follows:

1. Learning Motivation. Since students did not experience the role-play learning in the programming class before, their learning motivation was extremely high. Most of them paid attention to the role-play scenario and were eager to show their intellectual work. For example, students said, *Compared to traditional teaching (lecture), this (role-play activity) is fun and motivates us. The role-play scenario motivates us to spend more time on the activity. I am tired of lecture teaching. This activity just gives me energy to learn programming*.
2. Collaborative Learning. Through a group-based learning, acquiring skill in a learning programming language is productive rather than exhausting. Most students perceived that if the role-play activity is an individual assignment, disorientation or defeat may result. For example, students said, *Often, programming knowledge is complex.*
   I constantly need my classmate's de-bugging assistance. I cannot do it very well by myself. Knowledge in a two-people group is powerful. We can share what we know and fill the knowledge gap. Individual learning may get lost. I am not an active person. My classmate's strong involvement in the activity always pushes me to learn more.
3. Learning model. Transcripts produced a six-stage learning framework (see Figure 2) according to the characteristics of the role-play activity. The six stages are:

   a) Input: At this stage, each team received the instructional description of the role-play for a learning scenario. Most students were surprised that the programming class was implementing a new learning strategy. Students were willing to face a new learning challenge. For instance, students said, *We expected that the class would follow the traditional instruction from the beginning to the end. It really surprised me. I never thought that this type of class would use a new learning strategy.*

   b) Content Analysis: At this stage, each team began to analyse the role-play activity. Total understanding of the activity's description was the ultimate goal. For example, students said, *We want to clearly define what we want. If we do not understand the task, how can we design the programming questions? We are afraid that we will miss something important. So, we constantly ask the instructor about the activity requirements until we totally understand key points. We are unsure how to design programming questions. We constantly ask the instructor's opinions about questions' style.*

   c) Extensive Study: At this stage, each team extensively reviewed what they learned during the previous eight weeks. Team members shared what they knew in an effort to build a strong knowledge base. For instance, students said, *I think we should do an extensive review on programming knowledge before we engage in question design. We help each other to master the knowledge. We should achieve the same level (knowledge base). Otherwise, we cannot collaboratively design the questions. I know more than my team member. Before designing questions, I teach him several programming concepts.*

   d) Content Organisation: At this stage, each team began to design programming questions. Team members collaboratively discussed what content should be included. For example, students said, *We decided that each person should design five questions. Later, we combined all questions into a test to see if any question's content is similar. Like a jigsaw puzzle, we organised several programming concepts in one question. Sometimes, we found that some questions are very similar.*

   e) Content Testing: At this stage, team members collaboratively tested programming questions to see if any problems existed. For instance, students said, *Once we complete a question's design, we examined the feasibility in each question. We did not want other teams to identify any errors in our question items. It is like a debug process. We found that the item description in some questions is not logical. We modified these questions. Because the course instructor wants a high quality of programming questions, we did our best to polish test items. Sometimes, re-designing the question is a common task.*

   f) Output: At this stage, students submitted their tests to the course instructor and engaged in another role-play scenario. They all enjoyed solving test items that other teams designed. For instance, students said, *When we* *received the test, we found that test item to be simpler than ours. But the test solving still requires several programming concepts. We appreciate what other teams did. Their design thinking is different from us. We do learn a lot. Examining other teams' works is fun. After taking the test, we feel that we all read the content knowledge again, which allows us to achieve deep understanding of programming concepts.*

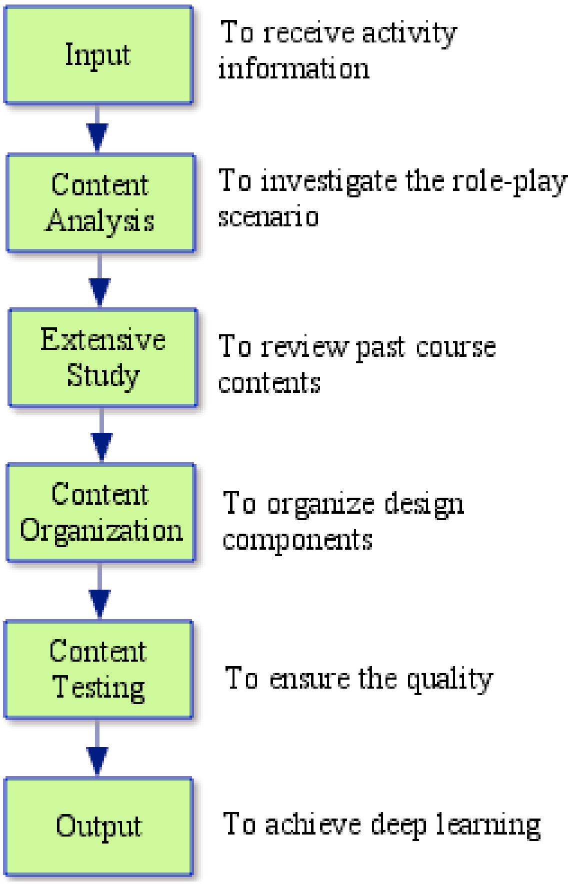

Figure 2: Role-play learning framework in this study.

CONCLUSIONS

From a quantitative perspective, the role-play learning strategy yields a significant effect on student learning according to the comparison between Class A (experimental group) and Class B (control group). In other words, if not considering extraneous variables, this study confirms that the role-play learning strategy can support engineering students in studying programming. Three possible explanations can interpret this phenomenon.

First, the design principles of the role-play activity were grounded in five theoretical foundations, which in turn greatly enhance the chance of producing significant results in an experiment. Second, students in Class A never experienced role-play learning activity in the programming courses before. High motivation, derived from new experiences, motivates students to achieve deep understanding of programming. Last, the qualitative results show that students engaging in the role-play activity extensively reviewed past course contents. Such a learning process facilitates knowledge acquisition and construction in students' minds [11].

From participants' perceptions of the role-play activity, a six-stage learning framework, including input, content analysis, extensive study, content organisation, content testing and output, was created. These six components support several theoretical concepts. For example, at the stage of extensive study, knowledge sharing between team members exhibits the importance of social interaction in supporting students' learning [12]; at the stage of content organisation and testing, creation of test items strengthens the concept of learning by doing and artefact-building [13].

Furthermore, the six components share similarities with five features in two learning models. For instance, content analysis is similar to *identification of facts* in the Problem-Based Learning (PBL) model [14] and *design an investigation* in Learning By Design (LBD) model [15]. Overall, the six components in the role-play learning model can enhance each feature in PBL and LBD.

Even though this study produced a significant result, potential improvements need further exploration. First, the course characteristics may influence the results of the study. Future studies can examine instructional effectiveness by applying the role-play learning strategy into different types of curricula. Second, implementing the role-play activity twice in the class may influence students' learning attitudes.

Future studies can observe the changes in students' motivation between two role-play activities. Third, instead of qualitative inquiry, future studies can employ a survey methodology to test the six-stage learning framework created in this study. Last, students' cultural backgrounds may influence their participation in the role-play activity. Finally, future studies can explore the influence of students' perceptions of cultural values on activity participation.

REFERENCES

1. Wilson, B.C., A study of factors promoting success in computer science including gender difference. *Computer Science Educ.*, 12, **2**, 141-164 (2002).
2. Mancy, R. and Reid, N., Aspects of cognitive style and programming. *Proc. 16th Workshop of the Psychology of Programming Interest Group*, Carlow, Ireland, 1-9 (2004).
3. Robins, A., Rountree, J. and Rountree, N., Learning and teaching programming: A review and discussion. *Computer Science Educ.*, 13, **2**, 137-172 (2003).
4. Hays, R.T., The Effectiveness of Instructional Games: A Literature Review. (2004-2005 Technical Report). Orlando, FL: Naval Air Warfare Center (2005).
5. Steinkuehler, C.A., Why game (culture) studies now? *Games and Culture*, 1, **1**, 97-101 (2006).
6. Mayo, M.J., Games for science and engineering education. *Communications of ACM*, 50, **7**, 30-35 (2007).
7. Foss, B.A., Game play in engineering education: Concepts and experiment results. *Inter. J. of Engng. Educ.*, 22, **5**, 1043-1052 (2006).
8. Ye, E., Liu, C. and Polack-Wahl, J.A., Enhancing software engineering education using teaching aids in 3-D online virtual worlds. *Proc. 37th ASEE/IEEE Frontiers in Education Conference*, Milwaukee, WI, T1A1-T1A5 (2007).
9. Van Manen, M., *Researching Lived Experience: Human Science for an Action Sensitive Pedagogy*. (2nd Edn), London, Ontario: The Althouse Press (1997).
10. Moustakas, C., *Phenomenological Research Methods*. Thousand Oaks, CA: Sage (1994).
11. Bransford, J.D., Brown, A.L. and Cocking R.R. (Eds), *How People Learn: Brain, Mind, Experience, and School*. Washington, DC: National Academy Press (2004).
12. Krajcik, J.S. and Blumenfeld, P.C., *Project-Based Learning*. In: Sawyer R.K. (Ed), The Cambridge Handbook of the Learning Sciences. New York: Cambridge University Press, 317-333 (2006).
13. Kafai, Y. and Resnick, M. (Eds), *Constructionism in Practice: Designing, Thinking, and Learning in a Digital World*. Mahwah, NJ: Lawrence Erlbaum Associate (1996).
14. Hmelo-Silver, C.E., Problem-based learning: What and how do students learn? *Educational Psychology Review*, 16, **3**, 235-266 (2004).
15. Kolodner, J., *Case-Based Reasoning*. In: Sawyer, R.K. (Ed), The Cambridge Handbook of the Learning Sciences. New York: Cambridge University Press, 225-242 (2006).

BIOGRAPHIES

Pao-Nan Chou is an assistant professor at the Department of Education at the National University of Tainan, Taiwan. He received his BS in Electronic Engineering and Computing Education and MS in Technological and Vocational Education from the National Taipei University of Technology, Taiwan. He also received his MEd and PhD in Instructional Systems from The Pennsylvania State University, USA. His research interests include e-learning and engineering education.

Hsi-Chi Hsiao is a Chair Professor in the Graduate Institute of Business Administration at Cheng Shiu University. He received his BS in Industrial Technology and Education from the National Taiwan Normal University, Taiwan. He also received his MEd in Industrial Education from The University of Wisconsin, USA, and his PhD in Curriculum and Instruction from The Indiana State University, USA. His research interests include human resource development and management, and engineering education.