

Effects of Program Visualization (Jeliot3) on Students' Performance and Attitudes towards Java programming

Nuttanont HONGWARITTORN

Department of Computer Science, Faculty of Science and Technology, Thammasat University
99 Mhu 18, Paholyothin Rd., Klong Neung, Klong Luang, Pathumthani, 12121 Thailand

and

Donyaprueth KRAIRIT

School of Management, Asian Institute of Technology
PO Box 4, Paholyothin Highway., Klong Luang, Pathumthani, 12120, Thailand

ABSTRACT

This research aims to examine whether the use of computer program visualization software could help students to improve their performance and have long-term effects on their attitudes towards objected-oriented programming (OOP). The computer program visualization software, Jeliot3, was used in this research as a learning tool for Java programming. The experiment was conducted with 54 students, divided equally into two Sections. All of them are studying in the information technology program at a private university in Thailand. The experimental results showed that, after controlling for the demographics and backgrounds of the students, the use of Jeliot3 could help students improve their learning performance in OOP and Java programming and obtain higher scores than those who did not use Jeliot3. However, it was found that the use of Jeliot3 did not affect students' long-term attitudes toward OOP and Java programming.

Keywords: program visualization, Jeliot3, object-oriented programming, performance, attitudes

1. INTRODUCTION

One major challenge in teaching programming in computer science and information technology programs is that learning programming is a time-consuming and practice-based process that takes long time for students to continuously focus on their learning process. The challenge becomes even more daunting when it comes to teaching a particularly complex language such as object-oriented programming (OOP). Learning OOP is a very difficult and complicated task [2], [5] because learners need to comprehend how to create objects and methods associated with such objects. Additionally, in contrast to structured programming, students usually found that learning OOP approach takes much more time and practice to at least start to fully comprehend its concepts. As a result, there have been continuous needs to conduct research and develop tools that can help learners learn and comprehend OOP easily and effectively.

2. LITERATURE REVIEW

Computer Program Visualization as a learning tool and Jeliot2000

Among several tools available to help the students learn how to program, computer program visualization is one widely accepted technique because it employs a graphical form to enhance users' comprehension of program execution or to visualize a process of data structure. Computer program visualization systems are not only created to enhance user's program comprehension, but also to help users locate or identify logical errors in the programs and facilitate them to analyse performance of program easily [15]. BlueJ is one of the famous program visualization systems that are widely experimented and used in teaching object-oriented programming by which students can see UML class diagrams and also they can interact with the classes and the class objects [8]. The software visualization tool, which could show a diagram of program or execution states, was tested to determine whether it would affect students' program comprehension. The test result revealed that the students who learned the programming class with such visualization tool could comprehend the object-oriented programming concept and write more correctible programs than those who learned the class without the tool [7].

Another widely used computer program visualization tool, Jeliot2000, was studied with two groups of high school students; the students in the experiment group saw the program demonstration and learned the programming labs through Jeliot2000, whereas those in the control group listened to the explanation of how the program works and saw the program demonstration from Turbo Pascal [10]. The analysis of students' assignments for the short-term effects and of the final exams for the long-term effects of Jeliot2000 indicated that the experiment group showed the improvement in their assignments and used more meaningful identifiers in their programs, but their performance seemed not to be affected by program animation in the long-term.

Moreover, the use of computer on learning of programming using software tools is not only limited to students but there have also been studies on instructors also. An example is the

paper titled “Perceived Behavior Control and Its Influence on the Adoption of Software Tools” [9]. This paper presents the results of a study of the attitudes that teachers possess on the use of the Jeliot animation system as a pedagogical tool.

Effects of Computer Usage as a learning tool on Students’ Attitudes

Most of the past research focused on how computer usage could be used as a learning tool to help improve students’ learning and comprehension, measured by performance such as scores after short-term usage. More recent studies started to pay attention to a long-term effect of such learning tools on the students’ attitudes [1]. Attitude has an important role on students’ behaviors. It indicates to a certain degree the possibility of adopting certain behaviors, for example, a new technology [3], [4]. However, there have been only few studies that attempted to measure students’ attitudes on complex subjects like OOP. Most of the previous studies focused on subjects like statistics, chemistry, and Western Civilization classes [1], [14]. The results of these studies on students’ attitudes are different due to different natures of the courses. Some studies found positive impacts of using computers as learning tools on students’ attitudes, while others did not. Nonetheless, most of the studies pointed to the same issue which is that students’ attitudes toward the subject can play significant role in students’ learning. As Gal, Ginsburg, and Schau (1997) noted, “students’ attitudes and beliefs can impede (or assist) learning statistics.”[1]. This statement is consistent with the statement raised by [3] above.

In this paper, students’ attitudes are defined as students’ perceptions toward OOP after learning OOP through a Java programming class.

Jeliot3 and its Effects on Students’ attitudes

After the release of the widely used Jeliot2000 program, Jeliot3 [13] was developed and released as a subsequent and improved version with the same purpose of serving as a tool to help improve students’ learning process. A paper titled “Jeliot3, an Extensible Tool for Program Visualization” discusses how Jeliot3 has been designed to allow for extensibility with modular design which would allow both original and other developers to add different visualization paradigms to Jeliot3, or to connect Jeliot3 with other environments. The capability of Jeliot3 to integrate with others system can ease its use by teachers and students [11], [12], [13]. It was expected that this new capability might help with the long-term students’ attitudes.

Nonetheless, quite a few studies on Jeliot3 were subsequently conducted in order to investigate if it could aid students in improving their learning performance. It was found that the program helped students in specifying the looping conditions, understanding the use of objects, and identifying sources of errors in the program. The results of several experiments were promising [6].

One study that clearly covered the aspect of Jeliot3 on students in an educational setting is the one by [11]. The study reports the preliminary findings of a qualitative investigation into how

students approach a program visualization tool, and whether the approach depends on how they are taught to use the tool.

These studies, however, did not measure the long-term effects on students’ attitudes. As a matter of fact, after the release of Jeliot3, there has never been any study on the attitudes of using Jeliot3 as a tool in learning OOP.

Therefore, the aims of this study are two-fold. The first and most important aim is to examine the long-term effects of Jeliot3 on student learning, measured by exam scores, and on students’ attitudes through the use of Jeliot3 in the Java programming class. The second aim is to develop a multi-dimensional measurement of students’ attitudes based on students in computer science and information technology majors in a private university in Thailand.

Academic Contributions

Based on the aforementioned sections, this study plans to study the under-investigated areas of using computer visualization tools by firstly developing a measurement for students’ attitudes on OOP. Secondly, the study hopes to use the measurement developed to investigate the long-term effects on students’ attitudes after using Jeliot3.

In order to achieve the objectives of this study, the following hypotheses are proposed:

H1: Students who learned Java programming with Jeliot3 would outperform, measured by their final exam scores, those who learned without any computer program visualization tools.

H2: Students who learned Java programming with Jeliot3 would have statistically different (positive) attitudes towards OOP, from those who learned without any computer visualization tools.

3. RESEARCH METHODOLOGY

This research is a quasi-experiment study with two groups, one experimental group which consists of students who learned Java programming with the program visualization tool, Jeliot3, and one control group which consists of students who learned programming without any computer program visualization tools.

Measurements

This study measures students’ learning performance and attitudes after 15 hours of lectures. The questionnaire to measure attitude was developed based on previous literature in computer science attitude survey [16]. It is a multi-dimensional measurement with the total of 44 items and was developed using five-step Likert scales (5 indicating strongest agreement and 1 indicating least agreement).

The overall Cronbach’s alpha value for the overall questionnaire items is 0.9887. The factors included in the questionnaire are confidence in own programming skills, attitudes toward programming, usefulness of having

programming skills, motivation to program, and attitudes for object-oriented programming. Cronbach's alpha value for each factor is above 0.81, which means that the measurement is very reliable.

The validity of the questionnaire was also verified by experts in the field of Human-computer interaction in Thailand and was accepted to be sufficiently valid.

TABLE I
Cronbach's Alpha for Constructs in the Questionnaire

Factors	Number of Items	Alpha Value
Confidence in own Programming Skills	11	0.8136
Attitudes toward Programming	6	0.9730
Usefulness of Having Programming Skills	9	0.9913
Motivation to Program	13	0.9967
Attitudes toward OOP	4	0.9967
Total	44	0.9887

Participants

There were total of 54 participants in this study. They are undergraduate students at a private university in Thailand. All of them are studying in the Information Technology Program and are majoring in either computer science or information technology. They are considered to have comparable backgrounds in computer and programming skills because they had to successfully passed the same courses of prerequisites in order to register for the subject, IT110 (Object-oriented programming), used in this study.

The subject IT110 used in this study was conducted in two Sections, each with 27 students, throughout the semester. This subject was offered in the morning and afternoon sections. Students registered this subject section depending on their study schedule. The students who registered in the first Section were assigned as a control group and those who registered in the second Section were assigned to be in the experiment group.

The questionnaires were distributed to all students for them to fill in before they left the exam room. The response rate is therefore 100%. Of all the 54 students who answered the questionnaires, 63% are female, 33.3% are male and 3.7% did not answer this question. Ages of the students range from 20 to 25, with a mean of 21.37 and a standard deviation of 1.14. The overall students' cumulative grade point averages (cGPA) before taking this subject range from 1.22 to 3.90, with a mean of 2.54 and standard deviation of 0.53. Results of the t-tests on the cGPA means between the two Sections indicated that students in the first Section have a higher cGPA mean than students in the second Section. The difference in their cGPA means is statistically significant at the 99% CI level. Except for the difference in the cGPA means, the two groups are otherwise statistically comparable in terms of their demographics and other backgrounds with no other factors statistically significant

differ.

Procedures

Both sections of this subject were taught with the same instructor, the outline and all materials except the OOP programs explanation or demonstration. Each lecture period was three hours a week. The topics which were covered in five weeks were listed in the table II.

TABLE II
Topics covered within five week

Week	Topics
1	Primitive data types and operations - Identifier, variables, constants - Data types: Numeric, character, and Boolean - Operators and assignment statements - Control statements: selection, loop, break & continue
2	Method - Creating methods, calling methods, and parameter passing - Overloading methods - recursion
3	Object-oriented programming I - Defining classes - Creating objects - Constructors - Scope of variables
4	Object-oriented programming II - Superclass and subclass - Overriding methods - Polymorphism dynamic binding
5	String and array class - String - String tokenizer - Array

The lectures for the control group were in the regular classroom, in which the instructor demonstrated how OOP works and showed examples without any program visualizing tools. The lectures for the experiment group were given in the lab-room. The student computers were connected and controlled by the program NetOpSchool. The demonstration of OOP program execution and examples through Jeliot3 was broadcasted to all student screens. The lecture contents, assignments, and examples were the same for both groups.

The exams were given after 15 hours of lectures and lasted for an hour and forty minutes.

4. DISCUSSION OF RESULTS

After data collection and data entry processes, the SPSS statistical software was used to analyze the data. Multiple regression analysis was used to examine the data with the demographics (gender and age) and academic background (cGPA) data as control variables. The data passed the normality test and other criteria for linear regression.

Effects of Jeliot3 on Students' Performance

The first regression was run using students' exam scores as the dependent variable. The R-Squared value for the model is 0.134. This means that 13.4% of the students' exam scores could be explained by our model. The model is statistically significant at the 95% CI level. The only independent variable that is statistically significant (at the 95% CI level) in the model is the "Section Number" variable, indicating that there is a statistical difference in students' performance measured by exam scores between the students in different Sections. Since there were no other differences in this experiment that are not controlled for in the regression model, it can be concluded that the differences in the students' performance between the two Sections are due to the use of Jeliot3.

Based on the standardized coefficient of the model, the coefficient of the Section Number is 0.414, which means that the students in the second Section, the experimental group with Jeliot3, performed better than the control group with no use of Jeliot3 score-wise. It is worth noting that the students in the second Section were able to achieve higher exam scores through the use of Jeliot3 despite the fact that their cGPA mean is lower than the students in the first group and that the difference in their cGPA means was controlled for in the regression model.

The results of this regression suggested the acceptance of our proposed hypothesis H1 that Students who learned Java programming with Jeliot3 would outperform, measured by their exam scores, those who learned without any computer program visualization tools.

Effects of Jeliot3 on Students' Attitudes

The second regression was run using students' attitudes as the dependent variable. The model is not statistically significant at the 95% CI level. This means that no variables in the regression model can statistically predict students' attitudes toward OOP.

The fact that the model is not statistically significant also implies that the use of Jeliot3 does not have the long-run effects, especially when measured using students' attitudes.

The results of this regression suggested the rejection of our proposed hypothesis H2. Students who learned OOP through Java programming with Jeliot3 did not have statistically different attitudes towards OOP, from those who learned without any computer visualization tools.

Furthermore, it was found that there was no statistically significant correlation between students' attitudes and students' exam scores.

5. CONCLUSIONS

This research aimed to evaluate the effects of the use of computer program visualization software as an additional tool in teaching OOP, more specifically in this study Java programming, on students' performance and attitudes toward OOP.

The students were divided into two Sections which are comparable and have no statistically different demographics and academic backgrounds, except for their cGPA means, which were controlled for in the regression model. One group of students was treated with the use of Jeliot3 program to help with their learning while the other one was not. The environmental factors, such as the context of the lectures and examples were the same for both groups throughout the time of this study.

The first result of the study on the effects on Jeliot3 on students' performance supports the previous literature that the use of computer visualization program can help improve the students' learning performance. This finding was supported by the statistical results showing that the students' performance, measured by exam scores, in the group using Jeliot3 is higher than the other one that did not use Jeliot3, while other environmental factors were held constant or controlled for.

However, the second result of this study showed that the use of Jeliot3 did not produce any statistically significant difference in students' attitudes between the two groups toward programming OOP. Therefore, the use of computer visualization program as a learning tool cannot be claimed to have long-term effects on the students' attitudes in this case.

6. LIMITATIONS AND FURTHER STUDY

One limitation of this study is the numbers of students in each Section, which are only 27 in each Section. It is possible that with a larger class size, one would be able to achieve clearer results in terms of statistical analysis. However, the class size in this study was beyond the researchers' control because it depended solely on the number of students willing to enroll in the program and in each Section each year.

Another arguable factor is the students' cGPA means, which are statistically different between the two groups in this study. Here again is where the limitation of this study lies in because students got to choose the Section they wanted to register for and the researchers cannot control this process. However, the researchers tried to control for this effect by incorporating students' cGPA means as a control variable in the regression model.

In addition, the result of the experiment on students' exam scores actually showed that the difference in the cGPA means between the two groups did not affect the students' performance. This can be seen from the fact that students with higher cGPA means but with no use of Jeliot3 performed worse than students with lower cGPA means but use Jeliot3.

The last limitation of this study is the R-squared value which is still not very high, though acceptable, despite the fact that variables collected were entered into the regression model. This could imply that some factors that really affect the model were not included in this study. These factors could include variables such as subjective norms, eagerness to initially enroll in this program, and attitudes toward the lecturer. By incorporating these other variables into the regression model might help increase the R-squared value.

Further study could include the aforementioned recommendations and could expand on the cultural and psychological dimensions of the subjects in the study or on other subjects in OOP in order to generalize the results of this study.

In addition, this study focuses only on Jeliot3 since it is one of the most widely used visualization tool and is commonly used to teach OO programming. There are also other software such as Ville, JeCo, and Jive, which can also be used to teach programming and are not tied specifically to any specific languages. The comparative study on this software could provide interesting aspects to improve the future use of visualization software as a learning tool.

7. ACKNOWLEDGMENT

The authors would like to thank the Jeliot3 development team who allowed us to use Jeliot3 for our study and special thanks to Ms. Nualchan Kasemsuksomboon for the help in data collection, the lecturer, and students who participated in this experiment.

8. REFERENCES

- [1] Alldredge, J. Richard, H. Dean Johnson, and Jose J. Sanchez: Does Viewing Video Of Statistics In Action Affect Student Attitudes?, **Proceeding of the ICOTS-7**, (2006).
- [2] Bennedsen, J., Caspersen, M. E.: Teaching object-oriented programming-towards teaching a systematic programming Process: **Proceeding of the Eighth Workshop on Pedagogies and Tools for the Teaching and Learning of Object-Oriented Concepts**, Oslo, Norway, (2004).
- [3] Berteau, P.: Measuring Students' Attitude Towards E-Learning. A Case Study. **The 5th International Scientific Conference on eLearning and software for education (eLSE)**, Bucharest, (2009).
- [4] Despotakis, T. C., Palaigeorgiou, G. E., Tsoukalas, I. A.: Students' attitudes towards animated demonstrations as computer learning tools.: **Educational Technology & Society**, vol. 10, no.1, 196-205, (2007).
- [5] Garner, S.: Learning resources and tools to aid novices learn programming: **Proceedings of Information Science + Information Technology Education Joint Conference**, Pori, Finland, pp. 0213-0222, (2003).
- [6] Kannusmäki, O., Moreno, A., Myller, N. Sutinen, E.: What a novice wants: Students using program visualization in distance programming course: **Proceedings of the Third Program Visualization Workshop (PVW'04)**, pp. 126-132, University of Warwick, United Kingdom (2004).
- [7] Kasmarik, K., Thurbon, J.: Experimental evaluation of a program visualization tool for use in computer science education: **Proceedings of the Asia-Pacific symposium on Information visualisation**, Adelaide, Australia, vol. 24, pp. 111-116, ACM Press (2006).
- [8] Kölling, M., Rosenberg, J.: BlueJ - The Hitch-Hikers guide to object orientation: Technical Report, 2, **The Maersk Mc-Kinney Moller Institute for Production Technology**, University of Southern Denmark, (2002).
- [9] Levy, R. B.-B., Ben-Ari, M.: Perceived Behavior Control and Its Influence on the Adoption of Software Tools, **ITiCSE'08**, Madrid, Spain, (2008).
- [10] Levy, R. B. -B., Ben-Ari, M., Uronen, P. A.: An extended experiment with Jeliot 2000: **Proceedings of the First International Program Visualization Workshop**, Porvoo, Finland, pp. 131 – 140, (2001).
- [11] Moreno, A. and Mike S. Joy, Jeliot3 in a Demanding Educational Setting, **Electronic Notes in Theoretical Computer Science**, 178, pp. 51-59, (2007).
- [12] Moreno, A., Myller, N. Bednarik, R.: Jeliot3, an Extensible Tool for Program Visualization: **Proceedings of the Koli Calling 2005: 5th Annual Finnish / Baltic Sea Conference on Computer Science Education**, November 17 - 20, (2005).
- [13] Moreno, A., Myller, N., Sutinen, E., Ben-Ari, M. : Visualizing programs with Jeliot3: **Proceedings of the International Working Conference on Advanced Visual Interfaces AVI**, Gallipoli (Lecce), Italy, pp. 373-376, ACM Press, (2004).
- [14] Njagi, Kageni, PhD., Clemson University (Kenya), Professor Ron Smith, Massachusetts Maritime Academy (USA), and Dr. Clint Isbell, Clemson University (USA): Assessing Students' Attitudes Towards Web-based Learning Resources, University of New Brunswick, www.unb.ca/naweb/proceedings/2003/PosterNjagiIsbell.html, (2003).
- [15] Oudshoorn, M. J., Widjaja, H., Ellershaw, S. K.: Aspects and taxonomy of program visualization: In: **Software visualization. Eades, P. and Zhang, K. (eds)**. Series on software engineering and knowledge engineering. London, World Scientific, 1, 7, (1996).
- [16] Wiebe, E. N., Williams, L., Yang, K., Miller, C.: Computer science attitude survey: Report No.: **NCSU CSC TR-2003-1**, Department of Computer Science, North Carolina State University, Raleigh, North Carolina, U.S.A. (2003).