

# Using Software Visualization In Learning Introductory Programming: A Review

Noor Fadzilah Ab Rahman, Nurkaliza Khalid,  
Azfi Zaidi Mohammad Sofi, Siti Zaharah Mohid

*Fakulti Sains dan Teknologi Maklumat  
Kolej Universiti Islam Antarabangsa Selangor (KUIS) Selangor,  
Selangor, Malaysia*

[noorfadzilah@kuis.edu.my](mailto:noorfadzilah@kuis.edu.my)

**Abstract.** The skill of programming includes knowledge of programming tools and languages, problem-solving skills, and effective strategies for program design and implementation. Basically, a common approach in learning programming is to be familiar with the basics of a programming language in advance to effective strategies for the whole programming process. However, many novice learners are facing difficulties and feel challenging in learning introductory programming course especially in JAVA programming language. Nowadays there are many technologies and programming tools are available to assist in reducing programming learning difficulties felt by novice learners. This paper provides a review of using software visualization as a programming tool in learning introductory programming focusing on understanding and behavior of learners.

**Keywords:** introductory programming, programming tool, programming learning

## INTRODUCTION

Programming is related to several fields of technology, and many university or college students are studying the basics of it. Unfortunately, many novice learners find programming a difficult thing to learn it especially in Java programming language. However, as one of the course objective, they are able to gain the skill of programming includes knowledge of programming tools and languages, problem-solving skills, and effective strategies for program design and implementation. Basically, a common approach in learning programming is to be familiar with the basics of a programming language in advance to effective strategies for the whole programming process [1]. In other words, they have to comprehend the fundamental of programming including flowchart, algorithm, syntax and structure of program.

Nowadays there are many technologies and programming tools are available to assist in reducing programming learning difficulties felt by novice learners. In fact, by using software visualization as a programming tool can help novice students to learn procedural and object oriented programming [2]. In this paper, we present

three selected software visualization tool utilized as learning tools for novice learner focusing on their understanding and behavior in learning introductory programming.

### Software Visualization

Software visualization represents many things to many people. In general, software visualization is the use of the crafts of typography, graphic design, animation and cinematography with modern human-computer interaction and computer graphics technology to facilitate both the human understanding and effective use of computer software [3,4].

This definition also includes different topics such as program visualization, algorithm animation, visual programming, programming by demonstration, data visualization, and source code browsers. This diversity is reflected in the taxonomic descriptions of the different field by researchers such as Price[3], Roman [5], Myers [6], and Stasko [7]. Software visualization refers to the visualization of computer program and algorithm [8]. Visualization tools can be very useful in teaching programming, primarily because their main purpose is to facilitate students' understanding of code execution by guiding them through a series of animated techniques [9]. The software visualization tools to be reviewed in this paper include Jeliot3, BlueJ, and VILLE.

### Jeliot 3

As in Figure 1, Jeliot3 is a program visualization tool that is based on the automatic animation of Java programs and it is used in tracing the execution of Java programs. As the execution advances step by step, the evaluations of expressions are visualized with graphical symbols. Jeliot 3 is designed mainly to support the learning process of novice programmers. Kannusmäki et al. [10] evaluated Jeliot 3 with qualitative methods and pointed out that only students without any previous programming skills were willing to use it. However, Jeliot 3 improved the novices' skills of perceiving if-statements and loops, understanding objects, and tracing errors from program code.

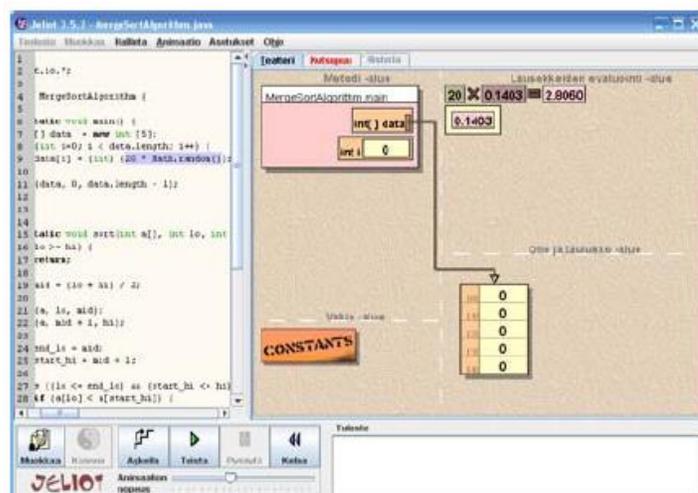


Figure 1: Jeliot3 environment

In research done by Cisar et al. [11], showed that students had achieved better results when they were using a software visualization tool, Jeliot3. Furthermore, their findings strongly suggested that Jeliot3 may yield the best results if implemented with students who are new to the art of programming. Meanwhile, a research by Hongwarittorn [12], found that the use of Jeliot3 did not affect students' long-term attitudes toward Java programming.

### **BlueJ**

BlueJ is an example of a static program visualization tool [13] shown in Figure 2. Unlike dynamic visualization tools such as Jeliot 3 and Ville, static tools which cannot visualize program execution step by step, but instead focus on visualizing program structure and the relations between program components. It places a special emphasis on interaction and visualization techniques to create an interactive environment that encourages exploration and experimentation with objects. Its wizards help learners create classes and implement interfaces. BlueJ has a class view showing relations between classes and an object dock containing all initialized objects.

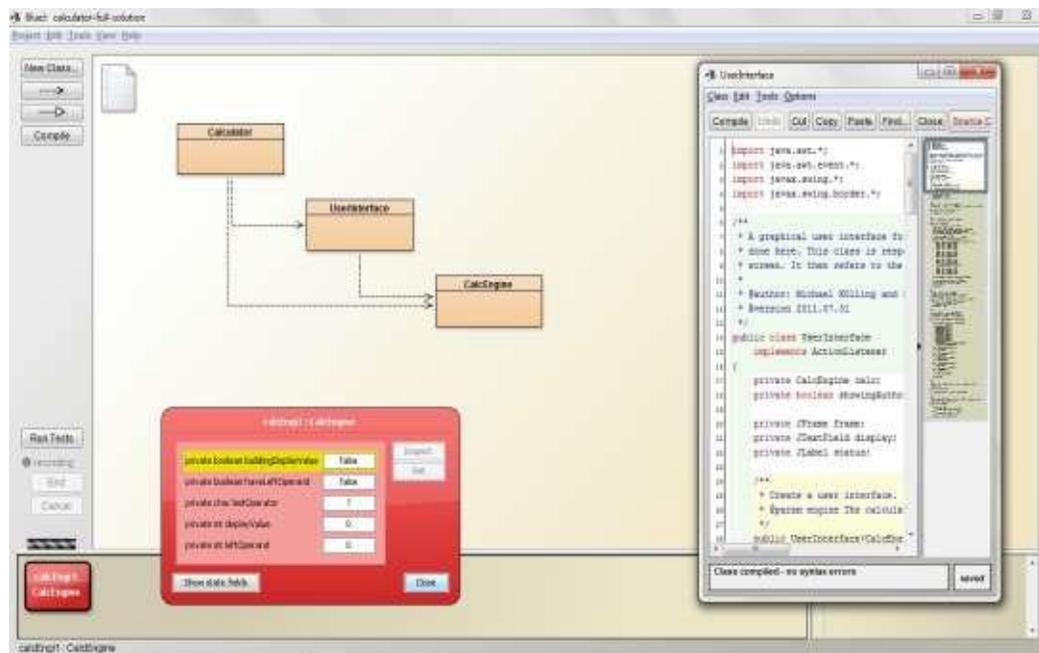


Figure 2: BlueJ environment

According to Kölling et al. [13], BlueJ is well suited to teaching programming with an objects-first approach. Dillon [14] found that visual tools could provide better understanding of programming to novice learner compared to command line tools. In addition, according to Kouznetsova [15], novices learners were able to develop their programming skills by using BlueJ environment. It is because the concepts of programming are presented in the context of a fun and familiar application. In addition, BlueJ was used by Uysal [16], Marceau [17], Kölling [18], Georgantaki [19], Van [20] in their studies.

## Ville

Ville is a program visualization tool, as showed in Figure 3 which can be used to create and edit programming examples and to observe events in the examples during their execution. Its main purpose is to support the learning process of novice programmers. It visualizes the execution of programs by displaying e.g. variable values, program line explanations and subprograms in their own areas.

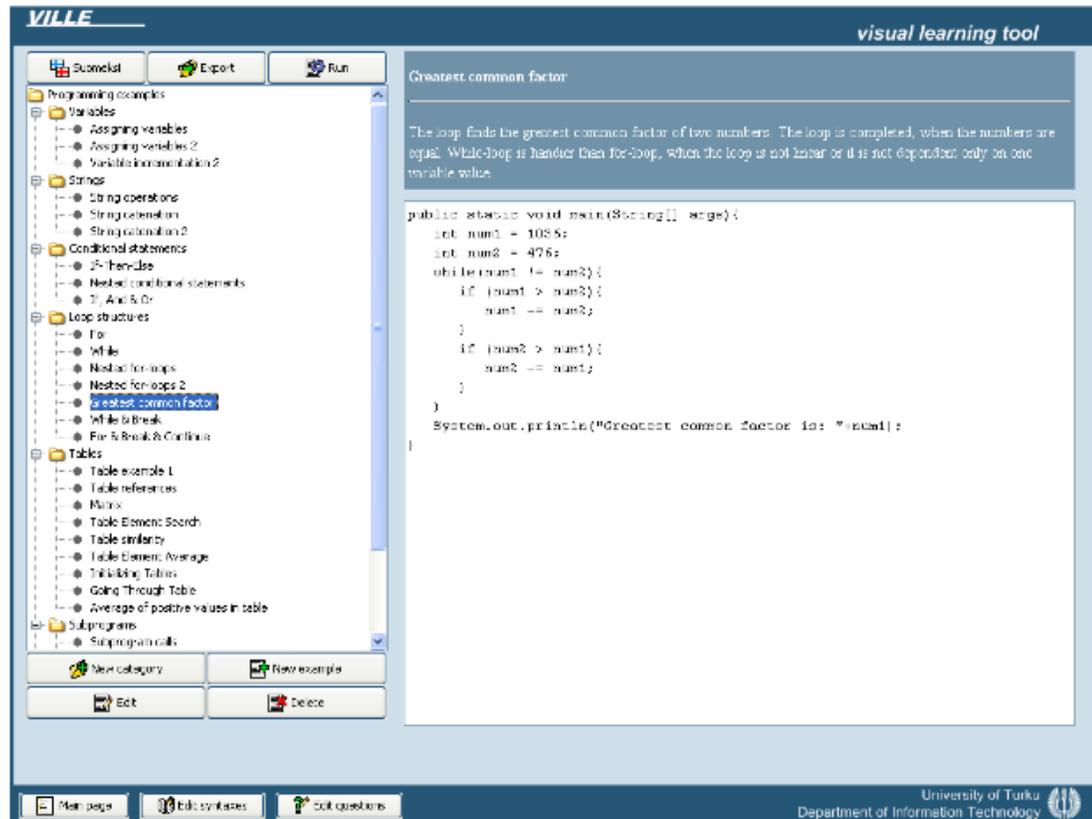


Figure 3: ViLLE environment

The program can be executed one step at a time, both forwards and backwards. The teacher can attach questions (multiple choice questions and graphical array questions) to fixed points of visualized programs. ViLLE supports a variety of programming languages, and comes accompanied with editors for defining new syntaxes, questions and exercises. In the study by Rajala [21] was found that ViLLE enhances the learning of students with no prior programming experience substantially, so that the differences between the novices and the more experienced learners were evened out as a result of a single training session. This indicates that program visualization indeed improves novice students' learning.

## CONCLUSION

This paper aims to review the use of three selected software visualization as a programming tool in learning introductory programming focusing in understanding the behavior of learners.

Based on related works, it showed that software visualization tool used to facilitate and enhance novice learners' learning in introductory programming in effective way. However, in this paper, we only review several software visualization tools. The chosen programming tools are Jeliot3, BlueJ and ViLLE. We noticed that each visualization tool have some advantages and disadvantages in their environment, features and main purposes of their development.

As a conclusion, further comparative study should be done on these software to provide interesting aspects to improve the future use of visualization software as a learning tool in introductory programming. Furthermore, determining the appropriate practices and programming tools for novice learner in learning programming has been an important research topic for some time.

## REFERENCES

1. Ala-Mutka, K. (2004). *Problems in learning and teaching programming-A literature study for developing visualizations in the Codewitz-Minerva Project. Codewitz Needs Analysis, 1-13. Retrieved January 22, 2012, from [www.cs.tut.fi/~edge/literature\\_study.pdf](http://www.cs.tut.fi/~edge/literature_study.pdf)*
2. Moreno, A., Myller, N., Sutinen, E., & Ben-Ari, M.(2004): Visualizing Programs with Jeliot 3. *Proceedings of the International Working Conference on Advanced Visual Interfaces AVI 2004.*
3. Price, B. A., Baecker, R. M., & Small, I. S.(1998), *An Introduction to Software Visualization.* England MIT Press, 4-26.
4. Price, B. A., Baecker, R. M., and Small, I. S.(1993), A Principled Taxonomy of Software Visualization, *Journal of Visual Languages and Computing*, vol. 4, no. 2, 211-266.
5. Roman, G.-C. & Cox, K. C.(1993), A Taxonomy of Program Visualization Systems, *IEEE Computer*, vol. 26, no. 12, 11-24.
6. Myers, B. A.(1990), Taxonomies of Visual Programming and Program Visualization, *Journal of Visual Languages and Computing*, vol. 1, no. 1, March 1990, 97-123.
7. Stasko, J. T. & Patterson, C.(1992), Understanding and Characterizing Software Visualization Systems, *Proceedings of IEEE Workshop on Visual Languages*, 3-10.
8. Maletic, J. I., Marcus, A., & Collard, M. L. (2002). A Task Oriented View of Software Visualization. In *Proceedings of the 1st International Workshop on Visualizing Software for Understanding and Analysis VISSOFT'02* (p. 32).
9. Hundhausen, C.D., Douglas, S.A. & Stasko, J.D. (2002). A Meta-study of Algorithm Visualization Effectiveness. *Journal of Visual Languages and Computing* 13:259-

290.

10. Kannusmäki, O., Moreno, A., Myller, N. & Sutinen, E. (2004). What a Novice Wants: Students Using Program Visualization in Distance Programming Course. In *Proceedings of the Third Program Visualization Workshop (PVW'04)*, Warwick, UK, 126-133.
11. Cisar, SanjaMaravic; Pinter, Robert; Radosav, Dragica; Cisar, Petar (2011). Effectiveness of program visualization in learning java: a case study with Jeliot 3. *International Journal of Computers, Communications & Control*; Vol. 6 Issue 4, p669
12. N. Hongwarittorn & D. Krairit (2010). Effects of program visualization (Jeliot3) on students' performance and attitudes towards java programming. *The spring 8th International conference on Computing, Communication and Control Technologies*
13. Kölling, M., Quig, B., Patterson, A. & Rosenberg, J. (2003). The BlueJ system and its pedagogy. *Journal of Computer Science Education, Special issue on Learning and Teaching Object Technology*, **13**(4).
14. Dillon, E., Anderson-Herzog, M., & Brown, M. (2012, September). Studying the Novice's Perception of Visual Vs. Command Line Programming Tools in CS1. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Vol. 56, No. 1, pp. 605-609). SAGE Publications.
15. Kouznetsova, S. (2007). Using BlueJ and Blackjack to teach object-oriented design concepts in CS1. *Journal of Computing Sciences in Colleges*, 22(4), 49-55.
16. Uysal, M. P. (2014). Interviews With College Students: *College Teaching & Learning*, 11(2), 59-70.
17. Marceau, G., Fisler, K., & Krishnamurthi, S. (2011). Mind your language: on novices' interactions with error messages. In *Proceedings of the 10th SIGPLAN symposium on New ideas, new paradigms, and reflections on programming and software* (pp. 3-18). ACM.
18. Kölling, M. (2008). Using BlueJ to introduce programming. In *Reflections on the Teaching of Programming* (pp. 98-115). Springer Berlin Heidelberg.
19. Georgantaki, S., & Retalis, S. (2007). Using educational tools for teaching object oriented design and programming. *Journal of Information Technology Impact*, 7(2), 111-130.
20. Van Haaster, K., & Hagan, D. (2004). Teaching and learning with BlueJ: An evaluation of a pedagogical tool. In *Information Science+ Information Technology Education Joint Conference, Rockhampton, QLD, Australia* (pp. 455-470).
21. TeemuRajala, Mikko-JussiLaakso, Erkki Kaila, TapioSalakoski, Effectiveness of Program Visualization: A Case Study with the ViLLE Tool. *Journal of Information Technology Education: Innovations in Practice* 7, IIP 15-32, 2008.
22. Rajala, T., Laakso, M. J., Kaila, E., & Salakoski, T. (2007). VILLE: a language-independent program visualization tool. In *Proceedings of the Seventh Baltic Sea Conference on Computing Education Research-Volume 88* (pp. 151-159). Australian Computer Society, Inc..