Available online at www.sciencedirect.com

ScienceDirect

ELSEVIER

Electronic Notes in Theoretical Computer Science 178 (2007) 137–144

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Visualizations in Preparing for Programming Exercise Sessions

Tuukka Ahoniemi[1] Essi Lahtinen[2]

Institution of Software Systems
Tampere University Of Technology
Tampere, Finland

Abstract

Visualizations are widely researched and used in teaching but the results of their benefits in learning are vague. We introduce an experiment of using visualizations in learning introductory programming. The aim was to support students in their preparation for the exercise sessions by using visualizations. The students' preparation consists of two phases that both are supported: reviewing the subject and a homework assignment. Thus this is also a novel approach to using programming visualizations and integrating them to the course content.

The experiment shows positive results especially among the students with no prior programming experience and the students who consider the programming course challenging. We conclude that integrating the use of visualizations to students' preparation for exercise sessions leads to better learning, more meaningful studying, and ultimately to better preparation. Therefore we also suggest this as a possible way for integrating visualizations to the course.

Keywords: Computer Science Education, Programming, Visualizations, Novice Programmers

1 Introduction

The learning problems in programming are often connected to more advanced issues than individual concepts, so the learning materials and situations should also be directed to develop more advanced programming skills [5]. One of the biggest learning problems of the novice programmers is that they have to handle abstract concepts of which they do not have a concrete model in their everyday life [7]. Thus, providing interactive visualizations as extra material for the students is a good way to concretize the subject in the beginning.

The most common use of visualizations is demonstrating a code example as an illustrative visualization. We wanted to make students participate in the visualization and integrate the use of visualizations to the students' preparation and homework

[1] Email: tuukka.ahoniemi@tut.fi

[2] Email: essi.lahtinen@tut.fi

doi:10.1016/j.entcs.2007.01.043

assignments for their weekly exercise sessions. The effects of this approach were tested in a real learning situation by in-class tests.

2 Background

The research done on the field of visualizations has resulted in instructions on how to build visualizations so that they will be pedagogically as beneficial as possible. For instance, Naps et al. recommend that the visualizations should engage the student to participate in the visualization actively [6]. As possible ways to do this it is suggested, e.g., that the visualizations should enable the user to provide his own input for the program and that there should be an interactive prediction in the visualization tool [8]. To increase the interactivity of the visualizations they can also be built to support all six stages of cognitive development listed in Bloom's taxonomy [4].

Despite all these recommendations and ideas on how to improve visualizations, the reports on their usage are diverse. A wide study conducted by Hundhausen et al. states that it is more important how the visualizations are used than what their content is [3]. In an other publication, Hundhausen reports that visualizations can actually distract the students' attention away from the subject [2]. On the other hand according to Ben-Bassat Levy et al. visualizations benefit the students with learning problems. This was also our main interest of research [1].

3 The Experiment

This experiment took place on an introductory course for programming *(CS1)* in Tampere University of Technology. The prerequisites for the course are limited to only basic knowledge of computer literacy and it is the first programming course for the students. The programming language used on the course is C++. There are weekly lectures and exercise sessions. The students ought to complete a small homework assignment prior the exercise session. The homework assignment requires them to familiarize themselves with the basics of the new subject. This usually also means reviewing the content of the lectures with the course material.

The idea of visualizations was familiar to the students already before the experiment. We had supported the students' own studying by providing visualizations on the course web page. The printed course material contains web addresses of the visualization examples and the visualization tool – VIP [9] – was also demonstrated on a lecture.

The experiment took place on the fourth and the fifth week of the course. On the first week of the experiment (the fourth week of the course) the exercise sessions dealt with loop structures and on the second week arrays. These weeks were chosen because both of the subjects are typically difficult for novice students [5] and they are easy to visualize.

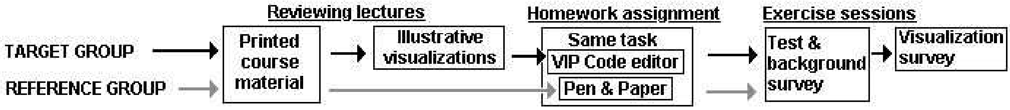


Fig. 1. Organization of the experiment.

3.1 The Method

We used two random groups of about 30 students who had enrolled for the excersise sessions. The target group used visualizations when preparing for the exercise session and the reference group did not. The organization of the experiment is illustrated in Figure 1.

3.1.1 Settings before the Exercise Session

Both groups had the printed course material for reviewing before the exercise sessions. Besides the printed course material, the students in the target group were provided an extra web page with instructions on how to review with the visualization examples and links to the examples. On both weeks the reviewing material contained two *illustrative visualizations* [4] to clarify the concepts.

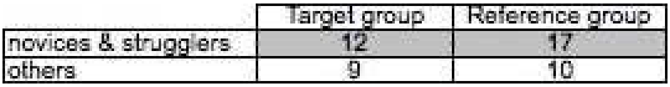
The actual homework assignments were exactly the same for both groups. The only difference was that the students worked on them using different tools. The reference group had the assignment available on the course web site. Most students in the reference group had used pen and paper to write the code and the answers to the questions. Some of them had also used a regular code editor and a compiler.

The web page provided for the target group contained the homework assignment as text just like for the other students. In addition, there was a link to a visualization tool where the student could start working on the code. VIP [9] contains a code editor where the student can write his own solutions, compile them and run them as a visualization.

3.1.2 Settings in the Exercise Session

On the experiment weeks, there was a short written test in the beginning of the exercise sessions to measure the students' learning. The students were not notified about the test in advance. They were not allowed to look at the materials and they returned their answers anonymously. The task was to write really small programs similar to the ones they had implemented in their homework assignments. The time was limited to only five minutes because the tasks tested the very basics and therefore would have been easily implemented in the time – assuming the subject was well learnt. We also wanted to have more variation inside the groups by limiting the time. Only the best students would complete the whole test.

Besides the small test, all the students responded to a short survey for background information, e.g., about their previous programming experience and how they felt about their progress on the course. Also the amount of time used, both on reviewing the subject and on doing the actual homework assignment, was asked. The students in the target group also answered another survey concerning the use



	Target group	Reference group
novices & strugglers	12	17
others	9	10

Fig. 2. The focused subset (highlighted with grey) was the novices and the strugglers of both groups. The amounts of students are shown in the table.

of the visualizations as a supporting tool for exercise preparation. The survey form was attached to the test so that the background information can be connected to the test answers.

3.2 The Homework Assignment

The exercise sessions in the first week dealt with loop structures. In the homework assignment there was a simple example of a `while`-loop. The task was first to find out what the piece of code does and to understand how it works. Then the students had to modify the code to implement an other kind of a functionality. The task reaches the level application (3) of Bloom's taxonomy of cognitive development, since it requires ability to apply one's knowledge in a new situation. Thus the assignment version implemented in the visualization tool is a *utilizing visualization* [4].

The subject in the second week was arrays. To widen the perspective of visualizational aid we chose this homework assignment differently: The students familiarized themselves with a given complex loop structure handling two arrays and answered questions related to it. The task requires identifying and analyzing the components of the code, so it is on the level analysis (4) of Bloom's taxonomy. Thus the version implemented in VIP is an *analyzable visualization* [4].

4 Results

On the first week, there were 21 students present in the exercise session of the target group and 27 students in the reference group, i.e. alltogether 48 students. On the next week the corresponding numbers are 21, 22 and 43.

As visualizations are mainly targeted for the novices and the students who have learning difficulties, we constricted the comparison of the groups to only the novices (no previous programming experience) or the ones finding the course subjects so far difficult or very difficult (here called *the strugglers*). The division and the numbers of the students in the groups is illustrated in Figure 2.

The results are divided into two parts: the effects on learning results and the effects on studying behaviour. The first represents the students' knowledge on the subject measured in the test as the second represents how the students prepared for the exercise session.

According to the results from the first week, the use of visualizations benefits learning: we found a statistically significant difference of the mean values of the test grades getween the groups. The results from the second week are analogous and support the results from the first week. Because of the smaller difference in the second week, this section mainly concentrates on representing the results from the

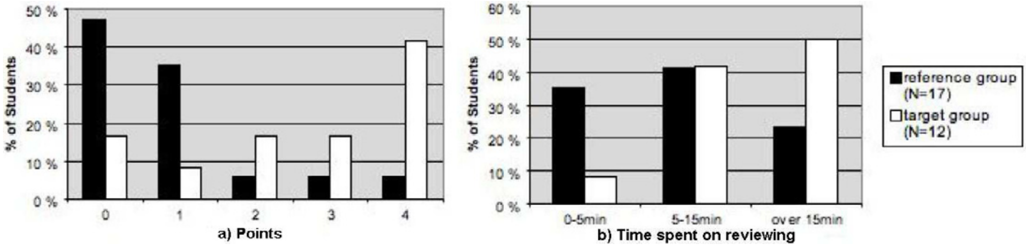


Fig. 3. Results from the first week of the experiment concerning the novices and struglers: (a) Distribution of the grades of the first task of the test and (b) Time spent on reviewing.

first week.

4.1 The Effects on Learning Results

The effects on learning results were analyzed by rating the students' answers for the test. For example, on the first week all three tasks were graded on a linear scale with points from 0 to 4 resulting the maximum of 12 points. On the second week the maximum was only 8.

The loop tasks seemed to be difficult for the students to complete in the given five minutes. The mean result was altogether only 3.5 out of 12 points (standard deviation 2.5). An independent samples T-test was used to analyze the difference between the groups. The means for the focused subset of novices and strugglers are 3.6 points (standard deviation 2.2) for the target group and only 1.7 points (standard deviation 1.5) for the reference group. This shows a significant statistical difference ($p < 0.05$). Even if the comparison is done to the whole groups (instead of only the focused subset) there is a small analogous difference between the groups.

In the next week, the corresponding means of the novices and strugglers are 3.1 points out of 8 points (standard deviation 2.3) for the students in the target group and 2.3 points (standard deviation 1.9) for the ones in the reference group. The trend is same as on the earlier week.

As the students carried out the tasks in the test sequentially, they all started with the first task. Figure 3a shows the percentage values of each grade in this task. Only the novices and the strugglers are taken into account. Almost all students in the target group (10 out of 12 = 83%) got at least one point and even 42% full 4 points as the reference group had the same numbers in 53% and 6%. The same phenomenon can be observed in the results of the second week.

4.2 The Effects on Studying Behaviour

Since the novices and the strugglers were the only ones whose learning results are different, it is logical that they are the only ones' whose studying behaviour was influenced by the visualizations. Thus this subsection concentrates only on the novices and strugglers of the groups.

According to the students' answers to the survey about their preparation, the students in the target group had used more time than the students in the reference group. Both the time spent on reviewing the subject and the time spent on doing the homework assignment were higher. The difference was bigger in reviewing the subject. The comparison between the time usage on reviewing the subject in the first week of the experiment is shown in Figure 3b.

More than a third of the students in the reference group spent less than 5 minutes in reviewing. More than 90% of the students in the target group spent longer than 5 minutes. It is clear that the students using the visualization tool concentrated longer even though the statistical significance between the groups can not be stated.

Also the feedback of the survey about visualizations as a preparation tool resulted in plain positive feedback. Students wrote comments like ”Though having read the specified course material, I really understood the subject after using the visualization examples.”

4.3 Comparing the Results of the Two Weeks

The experiment was not done in a strictly controlled situation but in a normal teaching group so some circumstances varied between the two weeks of the experiment. E.g., there were more absent students on the second week. The subjects on the two weeks were different so we also had a new type of homework assignment and a different test on the second week. All of these factors have influenced the results.

On the first week, the homework assignment was a *utilizing visualization* and on the second week an *analyzable visualization.* One important reason for the difference in the results can be that *utilizing visualizations* engage the student to produce his own code where as *analyzable visualizations* engage the student to observe the code intensively. The test performed in the class room was about producing their own code. So on the first week the preparation and the test were more similar than on the second week.

The in-class test was not announced in advance so on the first week of the experiment no one expected it. On the second week the students might have assumed that there could be a test again. Thus the students may have prepared better for the exercise session. This can also be one of the reasons why the statistical difference was not achieved on the second week.

5 Discussion

Even if the circumstances between the weeks of the experiment varied, it is advantageous that the experiment was done in a real learning situation. We captured the students’ experiences in a situation where they act as they would act normally when studying. Thus the results can better be applied to planning teaching in the future.

The results show that the use of visualizations helped the students who have most challenges in learning programming (the novices and the strugglers). They learnt more if they used visualizations when preparing for the exercise sessions. The students who had earlier experience in programming already had a mental model about the subject and thus the use of visualizations was not so helpful. Also the students who felt that the subject was easy could form the mental model without using visual materials. Hence, they did not benefit of the use of visualizations so much either.

Another result was that the students who used visualization examples along with the normal course material spent more time on reviewing the subject than the others. Studying obviously became more interesting as a new visual perspective was provided.

So what really can be concluded from the results is that visualizations do aid learning, but it is not sure whether this results directly of their usage. It can also result from the fact that when using visualizations, the studying itself is more interesting and the students use more time on it and thus learn better. However, it is not important, if the visualizations improve the learning results directly. The most important result is that they do improve them.

The difference between the two weeks of the experiment – the week when the students did a *utilizing visualization* exercise and the week when they did an *analysable visualization* exercise – also supports the recommendation from Naps et al. that the visualization should engage the student to work actively [6]. *Utilizing visualization* makes the student produce their own code where as *analysable visualization* only makes them analyze code written by someone else. The engagement to the visualization is more intense with a *utilizing visualization.* Also the learning results from the week when the *utilizing visualization* was used are better.

Using visualizations in students' preparation for exercise sessions had definitely a positive outcome because of the better learning. The exercise sessions ran smoother because students were better prepared due to the increase in their motivation. This also shows that using visualizations in preparing for exercise sessions is a working way of integrating visualizations to the rest of the course content.

The problems and considerations of this kind of approach are technical issues and the time spent by the teacher. Implementing tasks with visualizations requires quite advanced tools that have to be available for every student. Also preparing the tasks with a visualization tool takes more effort from the teacher than without a visualization tool.

When planning new ways to use visualizations in a course the teacher should also bear in mind that not all want to use new kinds of learning tools. As the use of visualizations mainly benefit the novices and the strugglers, it can be annoying for the students that do not need it. Some of the students might not like visual learning style or just have their own idea on how to work. Thus we recommend that the use of visualization tools is optional.

6 Conclusions

Using program visualizations improve the learning of students with no earlier programming experience and the students who have difficulties in programming. We cannot say whether the better learning results originate from the pedagogical impact of the visualizations or from the fact that the visualizations made the students study for a longer time. Either way, using visualizations improved the students' learning and preparation for the exercise sessions which was the purpose. Therefore, we recommend both using visualizations in teaching and using the exercise sessions to integrate the visualizations to the other parts of the course.

References

[1] Ben-Bassat Levy, R., M. Ben-Ari and P. A. Uronen, *The Jeliot 2000 program animation system*, Computers & Education **40** (2003), pp. 1–15.

[2] Hundhausen, C. D., *Integrating algorithm visualization technology into an undergraduate algorithms course: Ethnographic studies of a social constructivist approach*, Computers & Education **39** (2002), pp. 237–260.

[3] Hundhausen, C. D., S. A. Douglas and J. T. Stasko, *A meta-study of algorithm visualization effectiveness.*, Journal of Visual Languages & Computing **13** (2002), pp. 259–290.

[4] Lahtinen, E. and T. Ahoniemi, *Visualizations to Support Programming on Different Levels of Cognitive Development*, Proceedings of The Fifth Koli Calling Conference on Computer Science Education (2005), pp. 87–94.

[5] Lahtinen, E., K. Ala-Mutka and H.-M. Järvinen, *A study of the difficulties of novice programmers*, ITiCSE 2005, Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (2005), pp. 14–18.

[6] Naps, T., G. Rössling, V. Almstrum, W. Dann, R. Fleischer, C. Hundhausen, A. Korhonen, L. Malmi, M. McNally, S. Rodger and J. Velazquez-Iturbide, *Exploring the role of visualization and engagement in computer science education*, SIGCSE Bulletin **35** (2003), pp. 131–152.

[7] Robins, A., J. Rountree and N. Rountree, *Learning and teaching programming: A review and discussion*, Computer Science Education **13** (2003), pp. 137–172.

[8] Rössling, G. and T. L. Naps, *A Testbed for Pedagogical Requirements in Algorithm Visualizations*, ITiCSE 2002, Proceedings of the 7th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (2002).

[9] Virtanen, A. T., E. Lahtinen and H.-M. Järvinen, *VIP, a visual interpreter for learning introductory programming with C++*, Proceedings of The Fifth Koli Calling Conference on Computer Science Education (2005), pp. 125–130.