USING PROGRAM VISUALISATION LEARNING OBJECTS WITH NON-MAJOR STUDENTS WITH DIFFERENT STUDY BACKGROUND

Esa Kujansuu
Tampere Polytechnic - University of Applied Sciences
Finland
esa.kujansuu@tamk.fi

ABSTRACT

To improve the progress of the students of basic program courses program visualisation learning objects are produced and used as learning material. In some institutions the students have different study backgrounds. Some come to the course with non-theoretical and some with theoretical studies behind them. In this paper the study is focused on differences in effects of the program visualisation learning material with theoretical and non-theoretical background. The effects are collected as results from the course points and final grades and also from the survey where students evaluated the different sets of learning materials used during the course.

INTRODUCTION

The students of basic programming courses usually do not make much progress (Kölling & Rosenberg, 1996). To improve the students progress the Codewitz Network (Codewitz webpage, 2006) is producing program visualisation learning objects for the programming courses. In this paper the focus is set on the differences in the effects of program visualisation learning objects on the students' course results when having a different study background. The study was organised on the same course in two years: In the first year students do not have the program visualisation learning objects as learning material available and in the second year they have the program visualisation learning objects available. The students study exactly the same parallel course in two separate groups. The division of the groups is made according to the study background. The effects of the program visualisation learning objects on the results are then analysed by the final course points and grades and activity of the students and also with a survey about all learning materials available held at the end of the course.

STUDY SETUP

The study was conducted in Tampere Polytechnic during two years 2004-2005 for electrical engineering students who are non-major students in programming. These electrical engineering students have one obligatory programming course included in their studies. This course is called Basics of Programming and the course covers the first steps of programming like variables, selection, loops, arrays and functions. The Basics of Programming course is timed in the second semester of the first year of their studies and the scale of the course is four credits. The course consists of lectures and lab exercises. Each student has two lecture hours and three lab hours per week.

Each year two groups of around 35 electrical engineering students are formed. The first group (Group A) of students has vocational school as their study background. The most of the students have many years of practical working experience behind, but in most cases the work hasn't included programming. The background of Group A is thus called as non-theoretical. The second group (Group B) consists of students who mostly come to Tampere Polytechnic straight after upper secondary school graduation. The upper secondary school education in Finland is usually almost only theoretical studies. The most of the students of this Group B have

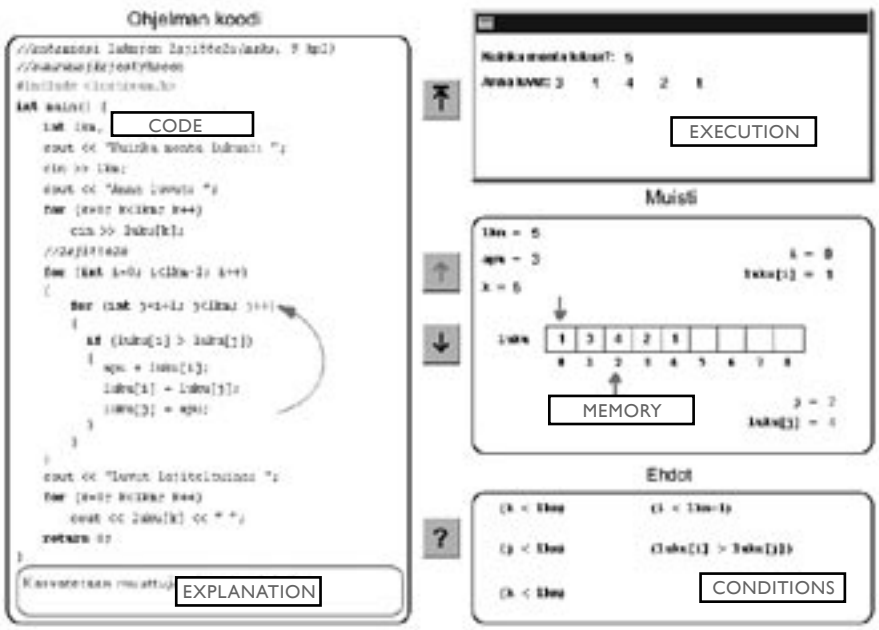
only a little or no working experience behind. Thus the background of Group B is called as theoretical. The both groups have the Basics of Programming course parallel with the exactly same content despite the differences in study background. It is assumed in the course everyone has no previous knowledge about programming.

The organisation of the study was divided in two years. In 2004 the students had the courses in a traditional way with no program visualisation learning objects available. The 2005 courses were organised with program visualisation learning objects available for the students as learning material in the local network throughout the course. In 2005 all the students were guided by the teacher to use the learning objects as the extra learning material. The program visualisation learning objects were occasionally also used as program examples by the teacher. The both year courses were held by the same teacher with the same material and with the same outside classroom assessments for the students. At the end exactly the same paper exam was given to the students in both years. The exam papers were not given back to the students. Thus the questions in the exam are considered not to be known by the next year students. There were 47 program visualisation learning objects available for the students during the course. The program visualisation learning objects were available for the students in their native language, which is Finnish in this study. The local network is accessible to the students also from home. Thus the program visualisation learning objects were available for them also when studying home.

This study consists of two parts. The first one is the results of the courses in two years. The first study takes also into account also the activity of the outside classroom assessments made by the students during the course. The second study is the survey made for the students at the end of the course. In this survey the students answered to the questions concerning their background and the usefulness of different kind of learning materials in their learning process. This survey was conducted during the last lessons of the course.

PROGRAM VISUALISATION LEARNING OBJECTS

Figure 1: The Active Areas of Program Visualisation Learning Object



The program visualisation learning objects (LOs) are defined in Codewitz network as follows: The program visualisation learning objects are browser capable, stand-alone, reusable, not linked to any other learning objects or resource and they are focused on one specific learning goal (Codewitz webpage, 2006). The LOs used in these courses were made in Codewitz Network during 2000-2004. The LOs used in this study were based both on program visualisation and exercise style. The major part of the LOs available in this study was the program visualisations and this study is concentrating only on these LOs. The program visualisation LOs covered all the areas of the course. The exercise style LOs covered only the loops and these LOs were introduced to the students by the teacher only when the loops were introduced.

The idea of the program visualisation learning objects is debugger like step-by-step program execution in both forward and backward directions (Figure 1.). The program code is highlighted in each important step of the program execution and the run of the execution in code is also visualised by arrows when necessary. In each step of the program execution console is visible as well as the memory area. There are also areas for the conditions and for the short explanations of the current step. The memory part is the only one where the layout can be changed according to the subject as learning goal. These changes appear for example in case of arrays when the structure of the array is visualised.

RESULTS

The course results:

The results consist of the final grade and the activity of the students. Final grade is between 0 and 5 where 0 is failed, 1 first grade for the completion of the course and 5 is the best grade. The activity of the students is measured according to the number of outside classroom assessments they have made during the course. The students got extra points according to the number of assessments made. At least 16 assessments made gave 1 extra point, 20 gave 2, 24 gave 3, 28 gave 4, 32 gave 5 and 36 or more gave maximum 6 extra points. The final grade was the sum of points from the exam and from the extra points. The number of students in the course results in Group A was 32 students in 2004 and 34 students in 2005. The same numbers in Group B were 36 in 2004 and 39 in 2005.

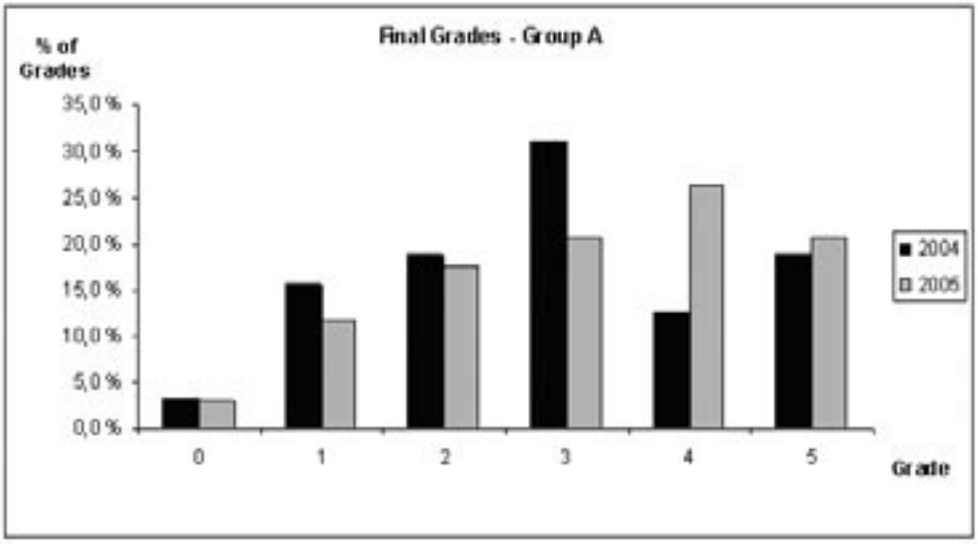


Figure 2. The Distribution of Grades in Group A

The activity of the students was slightly increased between the years in both groups. The activity of Group A increased 1 % and the activity of Group B increased 6 %. The more changes from 2004 groups to 2005 groups appeared in the number of students who gained no extra points. In 2004 in Group A there were 19% such students who gained no extra points. In 2005 in the Group A the same number was only 9 %. The same numbers in Group B were 17 % in 2004 and 13 % in 2005. In the area of activity of the students the rate was growing in both groups. The activity increased by 10 % in the Group A can be considered as significant improvement.

The number of good grades (4 and 5) increased in Group A significantly from 31 % in 2004 to 47 % in 2005 (Figure 2.). In the same group the number of passable or failed grades (0, 1 and 2) was decreased slightly from 38 % in 2004 to 32 % in 2005. In the Group B the number of good grades was slightly dropped from 50 % in 2004 to 46 % in 2005 (Figure 3.). The passable or the failed grade rate also dropped slightly in Group B from 33 % in 2004 to 31 % in 2005.

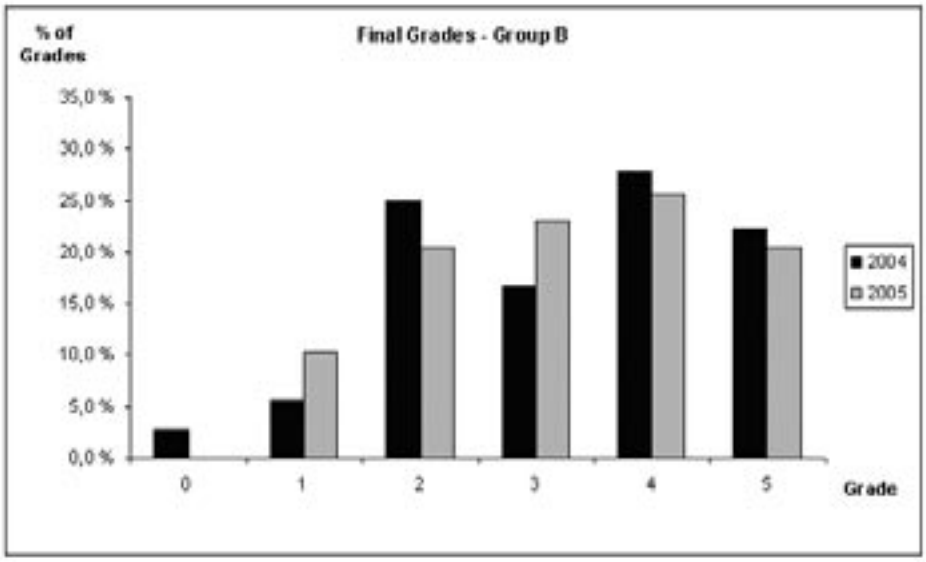


Figure 3. The Distribution of Grades in Group B

The survey results:

In the first part of the survey the students were asked about their programming skills before the course and about the skills in using computers in general. In both groups there where few students (5 %) who told they had average or good skills in programming before the course started. The most of the students who had any earlier experience in programming (94 %) estimated their skills as poor or very poor. Almost all the students (95 %) evaluated their general computer skills at least as average. In the area of background the students had no significant differences in background skills between the groups or between the years.

The students were asked about all learning materials used in the course. In Group A in year 2004 37 % of the students considered the lecture material to be useful or very useful. In 2005 in Group A only 15 % of the students thought the same way. In Group B in 2004 38 % of the students considered the lecture material to be needless or helping only rarely. In 2005 in Group B the same value was 33 %. Generally the lecture material was more valued in 2004 than in 2005. In Group A the change in the opinion of the lecture materials is remarkable. The exercises with the solutions were highly valued in both groups in both years. In 2004 the example programs were more valued than in 2005. In 2004 in Group A 95 % of the students considered the exercises

with the solutions to be useful or very useful when in 2005 the same value was only 73 %. In Group B in 2004 88% of the students considered the exercises with the solutions to be useful or very useful when in 2005 the same value was 77 %. In Group B the difference was less significant between years. The example programs were valued almost with the same rates with no significant differences when compared to the exercises with the solutions.

The program visualisation learning objects were used only in 2005 courses. This material valued to be at least sometimes helpful by 81 % of the students in Group A and by 70% in Group B (Figure 4.). The rate of answer "I don't know" was 23 % in Group B when in Group A it was only 8 %. In the Group A 46 % valued these learning objects as useful or very useful. In Group B the program visualisation learning objects were valued significantly more than in Group B. On both groups the program visualisation learning objects were not rated as the highest valued learning material. It seems though the existence of the program visualisation learning objects lowers the ratings of the other learning materials.

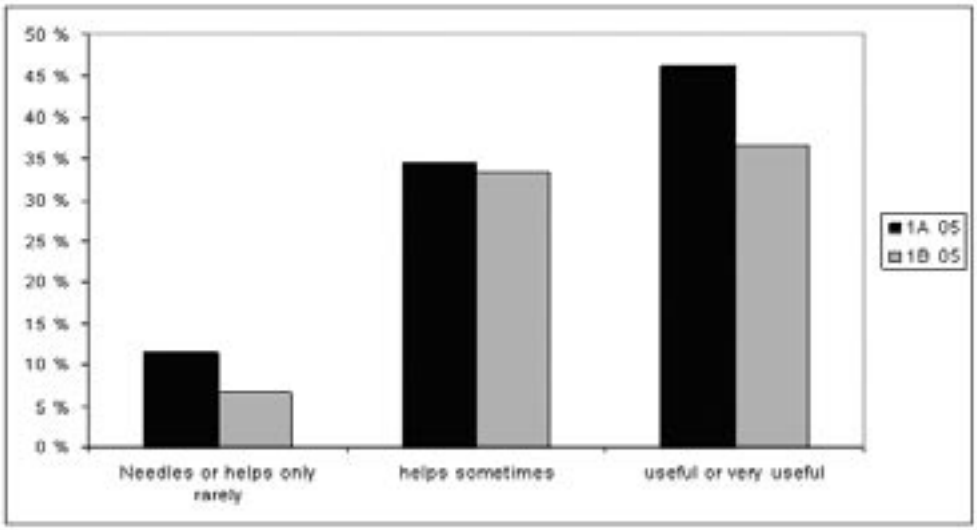


Figure 4. Evaluation of the Program Visualisation Learning Objects by the students

CONCLUSIONS AND FUTURE WORK

This study focused on the effects of the program visualisation learning objects when the study background is either theoretical or non-theoretical. According to the results the students having nontheoretical background like the program visualisation learning objects as learning material more. At the same time also the activity among these students was increased. It is not possible to say according to this study if the program visualisation learning objects had any role in the increment of the activity. The changes in course final grades from 2004 to 2005 in Group A leads to the conclusion the students were getting benefit from the program visualisation learning objects in their learning process. The students with non-theoretical background seem to get better grades with program visualisation learning objects than without such material. The students with theoretical background do not seem to get better grades even though they are also quite happy in using this kind of visualising learning material. When teaching programming to the students without theoretical studies behind them, it apperars to be possible to recommend the use visualisation as one part of the course material.

Since the amount of the students is not very large in this survey the next logical step will be to repeat the study with new groups of students. Future work will be directed to the learning process of programming

major students. One reason is the number of major students is larger in first programming courses. This will also give more ground for the results in this study. Also the study will be directed more into the course content issues. The important question for the future production of the program visualisation learning objects is to know in which content areas especially this kind of material helps the students and in which areas it does not help.

REFERENCES

Kölling M., Rosenberg J. (1996). Blue – A language for Teaching Object-Oriented Programming, *Proceedings of the 27th SIGCSE Technical Symposium on Computer Science Education*, pp. 190-194.

Codewitz project. (2004) *Codewitz Website* http://www.codewitz.net/, referenced 23.1.2006