13th International Educational Technology Conference

Teaching Programming in the Higher Education not for Engineering Students

Gabor KISS*

Óbuda University, Institute of Mechatronics and Vehicletechnics, Budapest, Hungary

Abstract

The algorithmic thinking is important in all professions, not just in the engineering work. My experiences show that the creating of the algorithmic thinking by the girls in secondary grammar school and in the higher education needs more time and other teaching methods. The numbers of the girls in non-engineering program is higher than in the engineering programs. I evolved a different (game-based) method to teach programming and to help them to acquire the algorithmic thinking easier for my non-engineering students. I thought programming for Economic information technology students with two colleagues. My colleagues used the conventional teaching methods that they use with engineering students in teaching programming and I used my evolved way and we had the same time to teach the learning material. At the end of the semester I separated the paper results of students by different teaching methods and compared. My starting hypothesis was that the group that I used my evolved teaching methods would achieve better results in the papers. Significance level was 5% through the analysis. Significant divergence in knowledge of students who took part on my lectures and the students who followed the conventional way was found. The students could get a one mark better paper results when they followed my method to learn programming and pass the test in higher percent. We can say my evolved teaching method is productive, and the students get better result when writing papers. At the beginning of the second semester more than 70% of students chose my course to learn the next level of programming.

* Corresponding author. Tel.: +36 30 9709 321; fax: +36 1 666 54 36.
E-mail address: kiss.gabor@bgk.uni-obuda.hu, gabor.kiss.phd@icloud.com
1. INTRODUCTION

Economic information technology students of the undergraduate course Programming I get acquainted with the basic programming knowledge (structured programming) in PHP. The Hungarian students do not have chance to learn programming before finishing the secondary grammar school and make the final exam (Kiss, 2012a; Kiss, 2012b), except for the students specialized in Informatics.

My earlier personal experiences in Higher Education showed that the majority of the students can not acquire the algorithmic thinking by following the traditional teaching process and they do not have too much chance to pass it. Unfortunately, I am not alone with this feeling. Three of the expert participants claimed that the difficulty in understanding the concept of programming and coding is because of the ineffective teaching strategies used during problem solving and coding (Ismail, Ngah, Umar, 2010). We can see all students have different levels of computer usage abilities by same Information Technology Education in other countries too (Isman, Celikli, 2009, Gastelú, 2013). It is the reason, that I think the creating of the algorithmic thinking by the undergraduate students need more time, or other teaching methods.

First of all we need to see how the human memory is working (Bloom, B.S.; Engelhart, M.D.; Furst, E.J.; Hill, W.H. and Krathwohl, D.R., 1956), the taxonomy of learning, teaching, assessing (Anderson, Krathwohl and Bloom, 2001) and the levels of learning to guide the students through the process of learning (Hoffmann, 2011). The performance of programming ability and thinking skills of students needs hard and concentrated work by teachers (Kurland, Pea, Clement, Mawby, 1986) and also subject Programming is important to attend the didactical methods of mathematic too (Ersoy, 2005).

Using playing games in teaching programming is productive (Esper, Foster, Griswold, 2013) and the kids can take the advantage of the different teaching methods. The situation does not change by students. My experience shows the LEGO-Mindstorm is a very good tool for learning programming, because the students can construct a robot with different functions and write programs without syntax error (Kiss, 2010a). This tool is useful by teaching programming for girls too and the half of the economic information technology students are girls, who have more problems to learn programming (Kiss, 2010b) and they are not so motivated to learn programming than the computer science engineering students.

The subject Programming requires the logical and algorithmic thinking from the students and the teachers have to develop these skills in the students, but the teachers have to consider the motivation and the precognition of the students before starting to teach any topic.

Although I would be useful for the economic information technology students, I could not use LEGO-Mindstorm for teaching programming. I have to follow the formal of the programming subject and teach PHP as a first programming language for these students.

I had an idea to write card games or dice games in my course. When the students could use the basic program elements like selection, iteration and can use the random generator in PHP after understanding the rules of the games, they could write a dice or a card games. The teacher can show the students how to use selection and iteration to build the ground for the game programs. For example the student play a dice game against the computer. The developed program generated randomly two (rolled) dice value, one for the student, one for the computer. We can choose a winner (who has higher value) by using selection. Another example can show how we can use the iteration combined with selection: the developed program generate randomly (rolled) dice values (more than one) for the two players (student and computer). The winner who has more rolled/generated “six” value. After this basic knowledge the students are able to combine these elements and write/develop complex game program with more rules.

We could not draw graphic for the game, we used just standard output opportunities to write the results after rolling the dices or taking a card and the computer was the second player with own decision progress. In this case the students could simulate a game against a computer. The students were very motivated to write different type of games (black jack, dice poker and so on) and they could understand easier how we can use the different
programming elements to write complex programs. The students were more interested in these examples than the traditional programming examples. I supposed those who learned programming by following this teaching method would have better paper results later. The economic information technology students attended the programming lectures held at the same time in two groups. In the first group (group A) of lectures (36 students) I made the course by using the game-oriented teaching methods while the courses for the second group (group B, 38 students) were followed the traditional way with normal examples from real life by my colleague who teaches programming for computer science engineering students at the university. The learning material was the same in the two groups, just the type of examples and exercises were different. Now I wanted to know the usefulness of all the examples and exercises I used.

My starting hypothesis was that the group where the students followed the game-oriented exercises would achieve better results in the papers.

After the semester I collected the paper results by groups and I tried to analyze whether this method was helpful or not for the students.

2. ANALYSING OF THE PAPER RESULTS

Some mathematical analysis was needed to decide whether using these game-oriented exercises were helpful or not in understanding the lectures and get better paper results.

2.1. The Number of Participants in the Tests in the Two Groups and the Values of Mean and Std. Deviation

According to the table (Table 1.) the mean of the results of papers in group A is higher. This group wrote the papers with a better result. It does not give enough information to state that the use of game-based programming education results in better written tests because this can happen accidentally, too. So, we needed more analyzing to keep the chance of accident low. I used the IBM SPSS Statistics v.20 by analyzing the paper results.

Table 1. Group statistics of the test results

<table>
<thead>
<tr>
<th>Group</th>
<th>Number of participants</th>
<th>Mean</th>
<th>Std. Deviation</th>
<th>Pass the test</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>36</td>
<td>2,72</td>
<td>1,28</td>
<td>82,9%</td>
</tr>
<tr>
<td>B</td>
<td>38</td>
<td>1,71</td>
<td>1,09</td>
<td>39,5%</td>
</tr>
</tbody>
</table>

If we spend more time looking at this table, we can see ~83% of economic information technology students who took part in the game-based programming education could pass the test and the students who used the traditional way of education passed the test in lower percent, but we still do not know if it is a coincidence.

2.2. Independent samples test

My null hypothesis was that the results of the paper written by the two groups of students would not differ significantly. Since we have two independent samples, we can use the independent sample test in SPSS to tell if the means of the paper of these groups differ or not (Table 2).
Table 2. Independent samples test of the test results

<table>
<thead>
<tr>
<th></th>
<th>Levene’s test for Equality of variances</th>
<th>T-test for equality of means</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F</td>
<td>Sig.</td>
</tr>
<tr>
<td>Equal variances assumed</td>
<td>1.55</td>
<td>0.28</td>
</tr>
<tr>
<td>Equal variances not assumed (Welch’s t-test)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

An analysis of the results of the economic information technology students showed, the variance of two groups are different, because the value of Levene’s test is not significant (p<0.05) (Levene, 1960).

In this case the means could be compared with T-test (Korpás, 2006). This showed difference between the means, because the value of T-test is significant (p<0.05). It means the use of the game-based programming education had influence on the results of papers of the economic information technology students.

2.3. Measures of Association by the Paper Results

Earlier, significant differences could be detected between the means of the papers written by the economic information technology students. It means it is profitable to make a deeper analysis to reveal the influence of the game-based lesson on the calculated means. I could reveal the influence with the calculation of the Eta-squared ($\eta^2$) (Cohen, 1973). The calculated value in percentage shows how much grouping influences the difference between means. Square root from the Eta-squared gives a value between 0 and 1 ($\eta$).

This shows the measures of association, i.e. how strong the connection between grouping and the achieved result is. The higher the value is, the stronger the connection is. In the next table we can see the calculated values and the strength of the connection (Table 3.).

<table>
<thead>
<tr>
<th>$\eta^2$</th>
<th>$\eta$</th>
<th>Strength of the association</th>
</tr>
</thead>
<tbody>
<tr>
<td>15.8%</td>
<td>0.40</td>
<td>middling weak connection</td>
</tr>
</tbody>
</table>

Calculating the Eta-squared, I tried to make the effect of the web-based education on the result of the papers written percentable and got ~16%. This means there is a middle weak correlation existing between using the game oriented programming education and the results of the papers written by the students. It seems the students could take advantage of using a game-based learning method before the test. We can remember the students who took part in the game-based lessons could pass the tests in higher percent (83% vs 40%).

3. Conclusion

After the analysing process, we can say my starting hypothesis is correct; students get better paper results by using a game oriented programming education method.

The economic information technology students could take advantage of this learning method tool before the test. The students could get a one mark better paper results when they took part in the game-based lessons and the number of the students who could pass the test ~doubled. The reason for this could be the fact that the game oriented lesson was preferred by the students who were motivated to write different type of dice or card game programs and passed the test in higher percentage. Before the students need to know and use it well the basic program elements like selection, iteration and so on. The traditional way to teach programming for economic
information technology students was not so successful than the game-based examples. The students were more motivated to write game programs and more students could pass the test. It means by same learning material and test exercises the student who learned programming in game-based method could have better paper results and pass the test in higher percent. The presented game-based teaching method is successful. The economic information technology students had to choose from two different courses to learn Programming 2 at the beginning of the next semester and my course was filled and closed in 3 minutes and the other was open days long.

Future work

One semester later the economic information technology students started to learn object oriented programming in JAVA, which needs other thinking method than the structured programming that the students learned in PHP.

It would be interesting to analyze the paper results to see the game-based learning methods working in this case too or not.

References

Esper, S., Foster R. S., Griswold G. W. (2013). On the nature of fires and how to spark them when you're not there. SIGCSE '13 Proceeding of the 44th ACM technical symposium on Computer science education (pp. 305-310)


Kiss, Gabor (2010a). Using the Lego-Mindstorm kit in German Computer Science Education. 8th IEEE International Symposium on Applied Machine Intelligence and Informatics (pp 101-104). IEEE Xplore digital library Digital Object Identifier: 10.1109/SAMI.2010.5423759


