International Conference NEW PERSPECTIVES in SCIENCE EDUCATION

Teaching Elementary School Pupils How to Code – Design and Delivery of a Programming Course

Tibi H. Moanes, PhD
Computer Science Department, Faculty of Education, Beit Berl College, Israel

Abstract
Elementary school pupils are interacting with mobile applications on a daily basis and are considered as digital native generation. Behind the development of such applications, there is a lot of hard work in programming that involves among other things, skills in programming and skills in algorithmic thinking. Learning programming is not easy for all students and neither for children. Programmers use mathematics, science, design, technology, and art to create their products. Thus, the understanding of the STEM concepts will help in designing new computer programs and applications. Furthermore, it is clear that in today's modern world coding is a basic skill and is especially important in STEM. In recent years, several academic institutions and companies have made much effort in developing web-based learning platforms to facilitate the teaching and learning of the foundations of computer science and programming. Most of these online programming environments are free, interactive and user-friendly. They have become easy to use, very attractive and helpful for teaching children how to code, during school and after school, but they are not based on comprehensive methodologies so that, the fundamentals of coding can be easily understood by pupils. The current research examined the teaching of a programming course to elementary school pupils, after school, based on the following three online interactive environments: "Plastelina", "Code with Anna and Elsa", and "Turtle Academy". The research included the development, implementation and evaluation of the course, which comprised of 13 meetings, total of 26 hours, taught to 42 pupils of 4th and 5th grades at two elementary schools in the center of Israel. The teaching and learning approaches that have been combined in this course were mainly game-, problem- and project-based approaches. In addition, the research aimed to investigate the pupils' attitudes toward the learning of computer programming, both before and after the participation in the course. Research data were collected by means of pre-post attitude questionnaire, written exam, and analysis of the final project and class observations. The findings indicated that the pupils' attitudes towards programming in general and towards motivation, competition, and challenge increased after the participation in the course.

Keywords: *Computer science education; elementary school; teaching and learning programming; teaching kids how to code*

1. Introduction
In today's world, most children learn to use smartphones and tablets before they learn how to read and write. Computers and other digital devices, as well as, digital learning tools are being used in almost every today's classroom. However, students in most elementary schools typically learn very little about what computers are made of, how they function, and what makes them work. Many parents and teachers are nowadays encouraging children to learn how to code so that they can understand how computers really works. A growing realization among educators is that teaching the principles of computer science and programming to children and school students will, not only make it easier for them to understand how information technology works, but it pushes them to think on a higher level, and be more creative and innovative [2], [4]. Creativity and innovation are crucial skills for future career success in all disciplines and especially in STEM.
Teaching kids how to code will let them look at a computer as a machine that carries out an algorithm. This in turn will help students to discover, examine and analyze new algorithms based on problem solving strategies and thereby develop their own problem solving, computational, and algorithmic thinking skills, which subsequently affect different aspects of their lives in a positive way. Wing [6] states that next to reading, writing, and arithmetic, computational thinking (CT) should be added to everyone's analytical ability. According to the relevant literature, the following five dimensions of CT can be identified: decomposition, generalization, abstraction, algorithm, and evaluation [2], [7], [9]. The application of CT is not limited to computers or computer science subjects. Denning [3] mentioned that CT has evolved from

being just the way that computer scientists think, to being useful in most other fields. Programmers use mathematics, science, design, technology, and art to create their products. Thus, the understanding of the STEM concepts will help in designing new computer programs and applications. Furthermore, teaching students, through coding, how to integrate technology into various disciplines, especially in STEM-subjects, would be easier.
In recent years, several academic institutions and companies have made much effort in developing web-based learning platforms to facilitate the teaching and learning of the foundations of computer science and programming. Most of these online programming platforms are free, interactive and user-friendly as well as supported with rich multimedia features. They have become easy to use, very attractive and helpful for teaching children how to code, during school and after school, but they are not based on comprehensive methodologies so that, the fundamentals of coding can be easily understood by pupils. The current research examined the development, implementation and evaluation of a programming course to elementary school pupils, after school, based on the following three online interactive environments: "Plastelina" for logic games, “Code with Anna and Elsa” via the Hour of Code project, and "Turtle Academy" for textual programming based on the Logo language.

2. The Study

2.1 The Study Plan and Objectives

The central axis of this study involved the development, implementation and evaluation of a course on computer programming principles for primary school pupils. The study also aimed to examine the impact of taking such a course on students' attitudes, also in terms of their interest and motivation, toward the learning of programming. Data were collected by means of pre-post attitude questionnaire, written exam, and analysis of a final mini project and class observations.

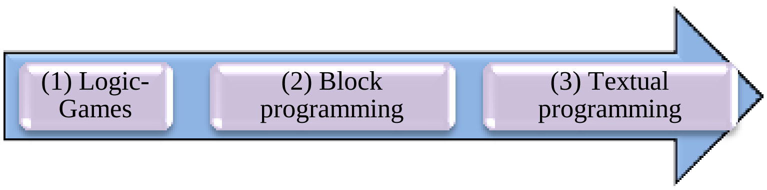
2.2 Research participants and settings

The study took place in two elementary schools located in two Arab cities in central Israel. The study population comprised of 42 pupils, half of them in 4th grade and the rest in 5th grade. The gender composition of the sample was 29 (69%) female and 13 (31%) male students. All of the participants were students at these schools. They were given a description of the project and its objectives and were invited to participate in the course. Most of the pupils had experience using online games and other devices such as Tablets and Smartphones.

2.3 Pedagogical Design

For this study, a programming course was developed which includes a sequence of three learning units involving three different learning environments. The environments were "Plastelina" for logic games, “Code with Anna and Elsa” via the Hour of Code project, which is a Blockly programming environment, and "Turtle Academy" for textual programming based on the Logo programming language. These environments were chosen because they are suitable for introducing the fundamentals of programming and the principles of computer science concepts to novice pupils. The main reason for using block-based tools before textual programming is to make programming learnable. In each unit, the pupils received handouts that include learning materials, examples and tasks. The three learning environments made up a sequence of three chapters, as illustrated in Figure 1.

Fig. 1. Three Learning Units Composing the Course Content



2.4 Teaching the Course

The main topics that were covered in the course included logical reasoning, computational and algorithmic thinking strategies, sequences, conditional statements, loops, and efficiency in coding. The course consisted of thirteen 90-minute sessions and included exercises, challenging tasks, and a final mini project. The pedagogy adapted in the teaching and learning of the course was mainly game-, problem-, and project-based while using STEM examples and problems that are relevant to students' interests and are appropriate for their age. Teaching and learning in the fields of science provides students an opportunity to develop increasingly important skills, such as problem solving, communication, and collaboration [8] as well as, to be creative and innovative [9]. Spiral learning, as well as, practical coding and programming based on active learning were also methods used during the teaching and learning of the course.

2.5 Methodology and Data Collection Tools

The survey instrument was a questionnaire that was carefully designed based on existing questionnaires [1], [5] in order to assess the impact of the course on students' attitudes. It consisted of 19 close-ended statements about attitudes towards learning programming. The statements were adapted for primary school pupils and were rated on a four-point Likert scale (strongly disagree, disagree, agree, strongly agree). They were divided into the following four categories: general attitudes, motivation, competition and challenge. The pupils filled out this questionnaire before and after the course. In addition to rating the survey statements, the pupils submitted assignments and a final project. Furthermore, qualitative data based on interviews and observations of the pupils' remarks and physical behaviours were also collected. Some sessions were also observed through video recording and imaging. This data was used to analyze, support, and explain some of the quantitative findings. Table 1 presents some items from the different categories.

Table 1. Some Items from the Questionnaire Categories

General Attitudes
- I would like to learn how to write computer programs, especially games. - Learning computer programming will help me in the future.
Motivation category
- I feel more excited when they asked me to write a difficult program. - When I write difficult computer programs, I do not feel fun and entertained.
Competition category
- I am willing to try hard to be the best in programming among my colleagues. - I try hard to write programs and solve difficult issues before the rest of my colleagues.
Challenge category
- I like to write computer programs that are challenging and need deep thinking. - If I am is required to write a difficult computer program I feel challenged and keep working on it until I finish it.

3. Data Analysis and Results

This study employed a quantitative data analysis. The collected data, about the pupils' attitudes towards the learning of programming in primary schools, with the help of a pre- and post- attitudes questionnaire was analyzed statistically using SPSS. Quantitative results revealed that pupils' motivation attitudes were higher, but not significantly, in the post-test (M=3.85, SD=0.38) than they were in the pre-test (M=3.74, SD=0.46). According to the challenging attitudes, the results showed that primary school students felt also little bit more challenged after taking the course (M=3.77, SD=0.47) than they were before the participation in the course (M=3.65, SD=0.53). Similar results were also found for the category competition. Means and standard deviations for all categories are listed in Table 2. General attitudes towards the learning of a coding course were found almost equal before and after the participation in the course. In all categories, no statistically significant differences were found between the mean scores of the pre- and post-questionnaire results. Table 3 presents some observations based on pupils' statements.

Table 2. Summary of the Main Findings Relating to Questions in the Pre and Post Tests (N=42)

		GA	MO	CO	CH
Pre Test	**Mean**	3.53	3.74	3.64	3.65
	SD	0.29	0.46	0.45	0.53
Post Test	**Mean**	3.56	3.85	3.75	3.77
	SD	0.21	0.38	0.55	0.47

GA= General Attitudes, MO= Motivation, CO=Competition, CH=Challenge

Table 3. Pupils' Observations

Observations
- Pupils continued to solve the exercises and did not go out for a break.
- They continued the exercise at home.
- They presented solution in several ways.
- They involved parents and friends in the learning.

4. Conclusions and Future Research

This study examined the development, implementation and evaluation of a programming course to elementary school pupils. The study also aimed to understand the pupils' attitudes toward the learning of computer programming, both before and after the participation in the course. Three different interactive coding environments were used during the course, while textual programming was introduces after block based programming. The results revealed, that the pupils' general attitudes towards programming remained positive also after the participation in the course. In addition, the attitudes of the pupils towards motivation, challenge, and competition increased after the participation in the course compared to the results received from the pre-questionnaire; however, the differences between the mean scores received from the pre- and post- questionnaire were not statistically significant. It can be concluded from this result, that pupils in elementary schools are capable of writing software programs using visual and textual programming environments, while it is recommended to start first with visual and then textual programming. Therefore, students in all elementary schools should be given the chance to learn programming at school in order to enable them develop the necessary skills of today's world. It is also recommended that the teaching and learning approaches that have been used in this study should be considered as the teaching and learning methods of any programming course. In future work, a larger sample size from different regions in the country should be considered, since it could provide sufficient variance among students and detect other significant findings. Moreover, further studies that investigate the impact of learning computer programming on the development of problem-solving, computational and algorithmic thinking skills are necessary.

References

[1] Baser, M. "Attitude, Gender and Achievement in Computer Programming", Middle-East Journal of Scientific Research, 14(2), 2013, 248-255.

[2] Crick, T. "Computing education: An overview of research in the field", 2017. Retrieved from https://royalsociety.org/-/media/policy/projects/computing-education/literature-review-overview-research-field.pdf

[3] Denning, P. J. "Remaining trouble spots with computational thinking", Communications of the ACM, 60(6), 2017, 33–39.

[4] Kaplancali, U. T., & Demirkol, Z. "Teaching Coding to Children: A Methodology for Kids 5+, International Journal of Elementary Education, 6(4), pp. 32-37, 2017. doi: 10.11648/j.ijeedu.20170604.11
[5] Morris, D., & Trushell, J. "Computer programming, ICT and gender in the classroom: a male-dominated domain or a female preserve?", Research in teacher education, 4(1), 2014, 4-9.
[6] Wing, J. M. "Computational thinking", Communications of the ACM, 49(3), 2006, 33–35.
[7] Wing, J. M. "Computational Thinking: What and Why?" The Link Newsletter, 2010. Retrieved from http://www.cs.cmu.edu/~CompThink/papers/TheLinkWing.pdf.
[8] Vahidy, J. "Technology and the curriculum: summer 2019", 2019. Retrieved from https://pressbooks.pub/techandcurr2019/chapter/enhancingstem/
[9] Palts, T., & Pedaste, M. "A Model for Developing Computational Thinking Skills", Informatics in Education, 19(1), 2020, 113-128. doi:10.15388/infedu.2020.06