

Primary School Pupils' Attitudes toward Learning Programming through Visual Interactive Environments

Khaled Asad¹, Moanis Tibi¹ & Jamal Raiyn^{1,*}

¹Computer Science Department, Al-Qasemi Academic College of Education, Baqa El-Gharbieh, Israel

*Correspondence: Computer Science Department, Al-Qasemi Academic College of Education, Baqa El-Gharbieh, Israel. E-mail: raiyn@qsm.ac.il

Received: August 31, 2016

Accepted: September 19, 2016

Online Published: October 9, 2016

doi:10.5430/wje.v6n5p20

URL: <http://dx.doi.org/10.5430/wje.v6n5p20>

Abstract

New generations are using and playing with mobile and computer applications extensively. These applications are the outcomes of programming work that involves skills, such as computational and algorithmic thinking. Learning programming is not easy for students children. In recent years, academic institutions like the Massachusetts Institute of Technology (MIT) and hi-tech companies, such as Google and Khan Academy, have introduced online environments to facilitate the teaching and learning of programming. Most of these programming environments are web-based, and interactive and are supported with visual multimedia features. Therefore, they have become easy to use, very attractive and helpful for teaching children how to program and to develop their algorithmic and computational thinking skills. The proposed presentation will describe research that examined the teaching of a course to primary school children based on three on-line interactive environments: "Plastelina" for logic games, "Code with Anna and Elsa" via the Hour of Code project block-oriented programming environment, for block programming and "Turtle Academy" for textual programming in the Logo language. The current research included the development, implementation and evaluation of the course at an elementary school. In addition, it was aimed at investigating the pupils' attitudes toward the learning of computer programming, both before and after participation in the course. The results revealed that the pupils' attitudes towards programming remained positive also after the participation in the course. It was also found that programming improved children's problem-solving skills.

Keywords: *computer science education; teaching and learning programming; visual programming environments; teaching children programming*

1. Introduction

This study introduces a learning strategy based on visual coding tools. According to various studies in visual learning theory, seventy-five percent of all information processed by the brain is derived from visual formats. Furthermore, visual information is mapped better in students' minds. The learner understands the visual code, when the learner sees information in a regular classroom (Williams, 2009). The above mentioned formats can be used by students to display large amounts of information in ways that are easy to understand and that help reveal relationships and patterns. Based on various studies, students remember information better when it is represented and learned both visually and verbally. These strategies help students of all ages better manage learning objectives and achieve academic success. Visual learning environments help students to understand information better and develop their visual thinking (Raiyn, 2016). Visual thinking is a learning style where the learner understands and retains information more effectively when ideas, words and concepts are associated with images. Visual information is designed with various interactive visual tools, such as information and communication technology (e.g., web services), 2D visual coding, and 3D visual environment. This study focused on interactive visual coding, such as "Code with Anna and Elsa" and "Turtle Academy", which uses the LOGO programming language. It addressed the question: To what extent can learning programming skills through the above interactive visual environments contribute to a positive attitude on the part of young students toward learning programming? The main goal of the current research was to answer this question.

2. Related Research

This section gives an overview of the use of visual tools in education. Rodger et al. (2009) introduced the integration of the Alice 3D into middle school and designed lessons in math, science, languages arts, social studies, and technology. Alice is an innovative 3D visual programming environment with which users can create interactive games, animations, and videos. It is a free teaching tool designed to create object-oriented programming. Ben-Ari (2012) introduced Scratch as a visual programming interactive medium in order to support the development of computational thinking skills. Wilson et al., (2009) introduced games-based learning, such as Scratch, and games-based construction into primary education to engage children at the primary level in developing computer programming skills. With Scratch, users can program interactive stories, games, and animations, which helps young people learn to think creatively. Stolee and Fristoe (2011) used Kodu Game Lab to introduce children to programming at an early age. Kodu is a 3D visual programming platform that can be used to teach creativity and problem solving, as well as programming. Ioannidou (2011) used games to support and develop computational thinking skills and to promote and increase opportunities for computer science education in the regular curriculum. Hero et.al. (2015) used the visual programming platform, MIT App, to increase interest and skills in computational practices. The visual programming platform, MIT App, enables users to create and design Android apps such as games, and more, and it can be used in various fields. MIT's App Inventor platform is used to learn how to program and create mobile apps. The materials available through the MIT App Inventor platform are suitable for teaching courses for middle school, high school, and college. Peluso and Sprechini (2012) used Alice to make a statement about the attitudes of high school student toward computer science. The students expressed their satisfaction with the use of Alice visual programming. The most common programming concepts gained from visual programming environments like Alice are loops (iteration), conditional statements that support problem solving, logic, reasoning and systematic thinking. We conclude that the development of educational visual programming environments like Alice, Scratch, Kodu, and Greenfoot, support the development of algorithmic thinking, and are effective for introducing primary school children to visual programming skills and concepts. Computational thinking includes logical and algorithm thinking which involves other kind of thought processes, such as, reasoning, pattern matching, and recursive thinking. (Wing, 2010). Consequently, this research aims at exploring students' attitudes towards learning programming. More specifically, the research was guided by the following questions:

- What is the impact of learning via visual and interactive environments in a programming course on students' attitudes towards learning programming at school?
- What is the effect of learning in a programming course on students' motivation to learn to use computers at school, and on their interest in pursuing a career in computers in the future?

3. The Study

• 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 aimed to examine the impact of such a course on students' attitudes in terms of their interest and motivation to learn programming at school and to pursue a computer related career in the future.

• Research participants and settings

The study will take place in a primary school located in a small town in northern Israel. The study population comprised 24 pupils, male and female, half of them in 4th grade and the rest in 5th grade. All of the participants were pupils at the school. They were given a description of the project and its objectives and were invited to participate. The gender composition of the sample was 4 male (16%) and 21 female (86%) students. Most of the pupils had experience using online games.

• Pedagogical Design

For the study, we developed a course on computer programming that includes a sequence of three different units involving three different learning environments. We choose these environments because they were suitable for introducing fundamental programming principles and concepts to novice pupils (primary and middle school students). In each unit, the pupils receive handouts that include learning materials, examples and tasks. The three learning environments made up a sequence of three chapters, as illustrated in Figure 1. Each student received lesson plans, lesson contents and descriptions and additional "tips and tricks". The three units in the three-week curriculum used visual interactive environment such as Plastelina, Anna and Elsa, and Turtle Academy.

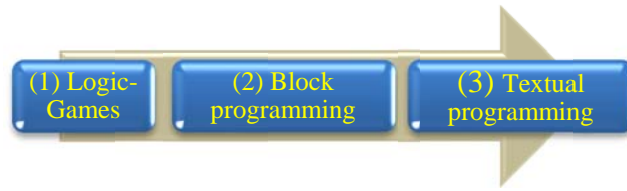


Figure 1. The Sequence of Three Learning Units Composing the Course Content

- The environment in the first unit consisted of the Plastelina Interactive Logic-Game. This environment included up to 15 different logic-games and was supported with multimedia features. All of the games were Web-based and presented problems that could be solved interactively. To solve them, the students had to use algorithmic thinking (see Figure 2).

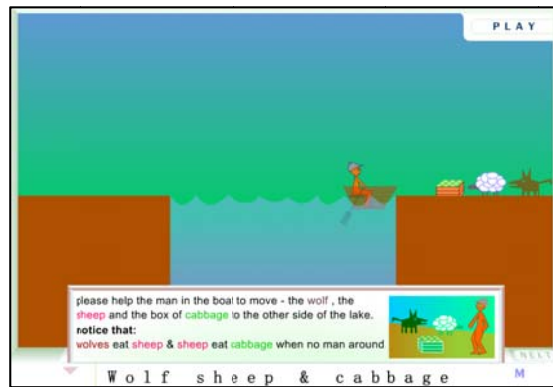


Figure 2. “Wolf, Sheep & Cabbage” an Interactive Logic Game

- The second environment was Code with Anna and Elsa, and could be reached via the Code website (code.org). This application presents a visual programming environment that introduces a series of programming tasks requiring the user to create programs using programming blocks (see Fig. 3).



Figure 3. Screen Shot of the Visual Environment of Code with Anna and Elsa

- The third environment used Turtle Academy, which works through the use of the LOGO programming language. It is an interactive and text-based programming that make it easy to write textual codes and create amazing shapes. The Turtle academy website offers a free textual programming environment using LOGO (see Fig. 4).



Figure 4. Screen Shot of the Textual Programming Environment of the Turtle Academy

• Teaching the Course

Two qualified undergraduates taught the course to the experimental group under the supervision of the main researcher. The course consisted of

Twelve 90-minute sessions and included the study of logic and algorithmic thinking, practical coding and programming based on active learning, exercises, and challenging tasks and projects.

• Methodology and Data Collection Tools

In order to assess the impact of the course on students' attitude, we use a half close-ended questionnaire. The three researchers developed the questionnaire using questions from Wiebe, Williams, and Yang (2003), adapted the questions for primary school pupils, and added questions based on the goals of the study. The pupils in the experimental group filled out this questionnaire before and after the course. In addition, notes were taken by the other teacher during and after each session. The notes were aimed at recording any special learning behaviors such as, enthusiasms for the tasks, dedication and concentration; they were used to analyze, support, and explain some of the quantitative results.

• Research tools

The survey instrument was a questionnaire that was carefully designed based on existing questionnaire (Baser, 2013; Morris and Trushell, 2014). It consisted of 29 questions about attitudes towards learning programming. In addition to answering the survey questions, the pupils submitted assignments and a final project. Furthermore, we collected qualitative data based on interviews and observations of the pupils' remarks and physical behaviors. We also made observations through video recording and imaging.

• Questionnaire design

The statements in the questionnaire were divided into three categories: motivation, competition and challenge. The questions were closed questions to be answered on a five-point Likert scale ranging from "strongly disagree" to "strongly agree". Table 1 presents some items from the different categories.

Table 1. Questionnaire Categories

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 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.

• Lesson Overview

Over the course of three weeks, as part of their ICT lessons pupils were given a one hour lesson involving the three online visual platforms for making games. The principle investigator led the lessons alongside the classroom teacher. During the sessions, the children solved interactive logical games, learned to build programs using visual programming blocks and wrote text programs using the Logo language. The pupils were asked to build/write basic and challenging programs and to complete a final programming project. Data were collected by means of pre and post attitude questionnaire, a written exam, an analysis of the final project and class observations. The experimental results indicate that teaching programming in a visual and interactive environments fosters children problem-solving skills. In addition, the presentation will report more results of the research, such as the students' achievements in learning programming skills in the course and the impact of the course on children's attitudes toward learning programming and computers.

4. Data Analysis

To collect the pupils' attitudes toward the teaching of programming in primary school, we have distributed a questionnaire. At this stage the SPSS was used to provide a descriptive analysis of the closed questions of the questionnaire. This study employed a qualitative data analysis. The closed questions of the questionnaires were analyzed statistically using MS Excel software. The descriptive statistics of the data were computed and to determine the frequencies of using programming in primary school, which formed the main research question. Note: In both categories, there was no significant difference between the pre and post tests. Tables 3 presents some observations based on pupils statements

Table 2. Summary of the Main Findings Relating to Questions in the Pre and Post Tests

	Pre Test		Post Test	
	MO	CH	MO	CH
Cronbach's Alpha	0.69	0.71	0.99	0.63
Mean	3.83	3.65	3.85	3.75
Std. Deviation	0.383	0.529	0.366	0.469
MO= Motivation				
CH=Challenge				

Table 3. Pupils' Observations

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

The graph in Figure 5 illustrates the fact that the motivation and the competitive spirit of the pupils were increased after they finished the course, whereas the perception of challenge in the pre and post tests quite the same.

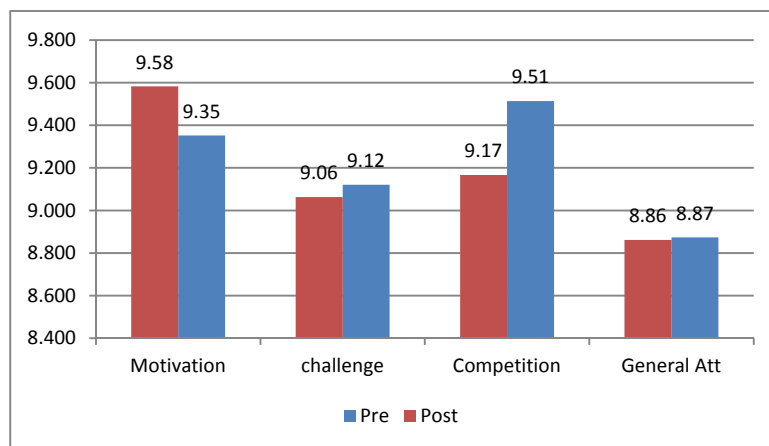


Figure 5. Main Categories of the Questionnaire

5. Conclusion and Future Works

This paper aimed to develop an exploratory model to explain and understand the pupils' attitudes toward the learning of computer programming, both before and after participation in the course. The results revealed, in general, that the pupils' attitudes towards programming remained positive also after the participation in the course. It was also found that programming improved children's problem-solving skills. This study was aimed at determining primary school pupils' attitudes toward the learning of programming with a visual, interactive platform. We found that self-efficacy, willingness, and the perception of the challenges presented by learning programming were high on both the pre and post tests. Furthermore, we conclude that pupils in primary schools are capable of writing software programs using visual code. Therefore, we propose that an academic unit based on visual interactive environments be taught in primary school. In future work, we plan to evaluate the impact of visual learning tools on the development of problem based learning (PBL). PBL has proven to be a successful educational strategy in various fields. It should promote the development of higher-order thinking (HOT) skills and collaborative skills in student.

References

- Baser, M. (2013). Attitude, Gender and Achievement in Computer Programming. *Middle-East Journal of Scientific Research*, 14(2), 248-255.
- Ben-Ari, M., & Resnick M. (2015). New frameworks for studying and assessing the development of computational thinking. *Computer Science Education*. Retrieved from http://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf
- Bednarz, Timothy F. (2011). *Developing Critical Thinking Skills: Pinpoint Leadership Skill Development Training Series*. Majorium Business Press.
- Brookhart, Susan M. (2010). *How to Assess Higher-Order Thinking Skills in Your Classroom*. Association for Supervision & Curriculum Development; 1 edition
- Cottrell, S. (2011). *Critical Thinking Skills: Developing Effective Analysis and Argument* (2nd edition). Palgrave Macmillan.
- Cottrell, S. (2013). *The Study Skills Handbook* (4th edition). Palgrave Macmillan.
- Herro, D., McCune-Gardner, C., & Boyer, M.D. (2015). Perceptions of Coding with MIT App Inventor: Pathways for their Future. *Journal for Computing Teachers*.
- Ioannidou, A. (2011). *Computational thinking patterns*. Annual Meeting of the American Educational Research Association (AERA).
- Macdonald, R., & Savin-Baden, M. (2004). A Briefing on Assessment in Problem-based Learning. *Learning and Teaching Support Network*. Retrieved from <ftp://www.bioscience.heacademy.ac.uk/Resources/gc/assess13.pdf>
- Morris, D., & Trushell, J. (2014). Computer programming, ICT and gender in the classroom: a male-dominated domain or a female preserve? *Research in teacher education*, 4(1), 4-9.

- Peluso, E.M., & Sprechini, G. (2012). The impact of *Alice* on the Attitudes of High School students Toward Computing. *Journal for Computing Teachers*, 7, 2012.
- Raiyn, J. (2016). Developing Mathematics Lesson Plan Based on Visual Learning Technology, *International Journal of Education and Management Engineering*, 6(4), 1-9. <http://dx.doi.org/10.5815/ijeme.2016.04.01>
- Rodger et. al. (2009). Engaging Middle School Teachers and Students with Alice in a Diverse Set of Subjects. SIGCSE09, March 3-7, 2009. <http://dx.doi.org/10.1145/1508865.1508967>
- Stolee, T.K., & Fristoe, T. (2011). Exprssing computer science concepts through Kodu Game Lab, SIGCSE'11, March 9-12, 2011, Dalla, Texas, USA.
- Wiebe, E. N., Williams L., Yang K., & Miller C. (2003). Report No.: NCSU CSC TR-2003-1 Dept. of Computer Science, NC State University, Raleigh, NC.
- Wilson, K.A., Bedwell, W.L., & Lazzara E.L. et al. (2009), Relationships between game attributes and learning outcomes: review and research proposals, *Simulation & Gaming*, 40(2), 217–266. <http://dx.doi.org/10.1177/1046878108321866>
- Williams, R. (2009). Visual Learning Theory. Retrieved from http://www.aweoregon.org/research_theory.html
- Wing, J. M. (2010). Computational Thinking: What and Why? *The Link Newsletter*. Retrieved from <http://www.cs.cmu.edu/~CompThink/papers.html>