# Perceptions of Informatics Teachers Regarding the Use of Block and Text Programming Environments

Margarita Karaliopoulou, Ioannis Apostolakis, and Evangelos Kanidis

Abstract-In this paper we investigate teachers' perceptions regarding the use of block and text programming environments in the class. The survey targets teachers of informatics in primary and secondary schools in Greece and attempts to answer research questions regarding the suggested duration of block-based programming practice and the difficulty of students' transition from block-based to text-based programming. In contrast to the majority of research works that consider students' opinions, in this paper we investigate the perceptions of their teachers and take advantage of their experience on the taught subject. Although the curriculum mandated by the Ministry of Education provides no specific directive, teachers agreed that block-based environments are appropriate introductory tools to programming. One of the primary tasks of this work was to determine the recommended age for students to move from block-based to text-based programming. The analysis of the collected data clearly indicated a specific age for this transition: teachers believe that students in primary school and the lower secondary school (ICSED levels 1 and 2 respectively) should use a block-based programming environment and should be introduced to textbased programming during the upper secondary school (ICSED level 3), after the age of 14. The findings of this study can be useful when designing new Informatics curricula for the secondary education, all-over the world.

*Index Terms*—Block-Based Programming; Text-Based Programming; Age Transition; Teachers' Perception; Curriculum.

## I. INTRODUCTION

Programming is a complicated and demanding task, which is difficult for young students and generally for novice learners to master. These difficulties arise from the fact that students should first learn to create algorithms to solve problems and then translate them into a programming language. Hence, they need to learn the vocabulary and syntax of a programming language to implement the solution to their assigned problem. Researchers have created graphical programming environments, designed to help novice programmers. The most common of these are Logolike environments using graphical user interfaces that facilitate programming. In this paper we distinguish between two types of programming environments: Block-based programming environments (Graphical User Interface / Drag and drop editing) and Text-based programming environments (Command User Interface / Text-based editing).

A common characteristic of block-based environments is the use of blocks of code, which are collated in order to create a program. Hence, students do not have to learn the syntax of a programming language and can focus instead in solving their assigned problem. The common characteristic of text-based environments is that students should learn the programming commands and type them correctly.

Block-based programming environments are designed to be used by novice programmers, mostly young students attending elementary or early secondary school grades. Programs created using blocks often reflect the syntax and structure of the underlying programming language. By using block-based programming students can create sophisticated programs in a multimedia context, with relatively little effort. Users can incorporate art, music and interactivity to create games, stories and applications [1]. Block-based programming is the most popular way of introducing students to programming, as it can help students develop their programming ideas in a non standard way [2].

Block-based programming environments are used worldwide as introductory tools. We report the case from the U.K. where environments such Scratch, Snap, App Inventor or Alice dominate the teaching of programming at elementary and lower secondary school classes [3]. "National curriculum efforts including Exploring Computer Science, the CS Principles Project and Code's curriculum materials, all utilize blocks-based tools to introduce students to programming" [4].

As block-based programming tends to dominate early school grades the questions arising are the following:

- 1) Does block-based programming make the transition to text-based programming easier?
- 2) How long should a student use block-based environments before transitioning to text-based programming?
- 3) Which is the appropriate age for this transition?

Most of these issues seem to remain open research questions, only recently being dealt by the research community. Most papers take into consideration students' opinions. In this paper we investigate the opinion of the teachers based on their classroom experience on the subject. This research was carried out in Greece. Although Informatics has existed as a school subject in Greece for many years, there is still no comprehensive Informatics curriculum spanning all grades.

In this paper, we first provide the theoretical background related to our research questions. Then we state the research questions and present our research results. Finally, we discuss our results and point out the most interesting

Published on February 2018

M.Karaliopoulou, Department of Mathematics National and Kapodistrian University of Athens Panepistimioupolis, GR-157 84, Athens Greece (e-mail: mkaraliop@math.uoa.gr).

I. Apostolakis is with the Ministry of Education, Research and Religious Affairs, Athens, Greece (e-mail: apost@sch.gr).

E. Kanidis is with the Ministry of Education, Research and Religious Affairs, Athens, Greece (e-mail: vkanidis@sch.gr).

findings.

## II. BACKGROUND

# A. Block-based programming, and its use in the classroom

Many researchers have studied the characteristics of block-based programming, when compared to text-based programming and have discussed its use in the classroom. Price and Barnes [1] concluded that novice programmers achieve better results in programming, when focusing on the end-task, instead of programming details. On the other hand, block-based programming environments do not enable writing and maintaining long and complicated programs, thus more mature programmers prefer text-based programming [5]. Many researchers have compared block to text-based programming [6], [4]. An overview of the bibliography, which discusses the advantages and disadvantages of block-based programming and a compares with text-based programming was recently published by Karaliopoulou and Kanidis [7].

Recent research has shown that although there is an increase in the popularity of block-based programming, there are still open questions concerning the pros and cons of its use in the classroom. Most papers are in favour of these environments for young students and non-typical teaching environment [8], [9].

During a recent conference (SIGSE'15), there was a panel discussion with the topic "Future Directions of Block-based Programming" [5]. The panel discussed, among others, the reasons why block-based programming is popular and some of the problems that arise when using it in the classroom. Weintrop and Wilensky [10] examined problems arising from the use of block programming in the classroom, especially in secondary education. They remark that some students do not consider block-based programming as "real" programming and regard its use in the classroom only for educational purposes. DiSalvo [11] has made similar remarks.

By conducting research among students, researchers have attempted to answer the question whether learning blockbased programming first, facilitates students' introduction to text-based programming later. Armoni et al. [12] concluded that previous student experience with Scratch in middle school helped them learn more advanced programming topics when using a text-based programming environment at the next educational level. However, at the end there was no significant difference in achieving the desired educational goals, between students with and without previous experience with Scratch. Similarly, Wolz et al. [13] observed that students, who had been taught Scratch even for a week, were more capable than others in transitioning to Java or C text-based programming.

Wagner et al. [14] asked students to repeat exercises using App Inventor first, a block-based environment and then using Java. The researchers concluded that students could reflect the procedures they had done with blocks to those procedures using text programming. Dorling and White [15] concluded that the combination of both types of programming environments, used in parallel, contributed to the students confidence in their abilities, independence and interest in learning a programming language in primary and secondary education. Kanidis, Karaliopoulou and Menounou [7] conducted research among students, who first had to solve a problem using block-based programming and then solve the same problem in a text-based environment, using a method called "Mediated Transfer of Knowledge" [16]. For example a program in Scratch can be reflected or compared command by command to the same program in Java [2]. They concluded that most students thought that their prior experience with block-based programming helped them in creating the program in the text- based environment.

The question arising is, how to decide which is the appropriate age to introduce students to text programming. The answer to this question may vary depending on the previous experience of the students: whether they have been taught block-based programming and for how long.

In the UK, the programming requirements of the national curriculum for children aged 7-11, are "typically met by block-based programming systems, with Scratch being by far the most popular". Students 11-14 years old "must use two or more programming languages, at least one of which is textual, to solve a variety of computational problems" [17]. "These requirements mean that students face a transition from blocks to text-based programming typically at an age of about 12 years old" [3]. Brown et al. [18] notice that "Given the current popularity of the Scratch programming system, this line is broadly interpretable as 'Scratch plus a text- based programming language'; while the requirements before age 11 can be fully satisfied in Scratch, the 11–14 curriculum deliberately ensures that pupils move to full-text programming".

Most papers take into account students' opinions. Saeli et al.[19] noticed that "usually curriculum designers leave the choice of the programming language to teachers and among secondary teachers appears to be heterogeneity in the choice of programming languages/ paradigm".

## B. Educational system in Greece

The Greek educational system is divided into three levels: primary, secondary and tertiary. The Primary/Elementary school, is called Dimotiko, lasts for six years and enrolls students aged between 6 and 12 years old. Secondary education has two levels: the lower level is called Gymnasium (Junior high school), lasts for three years (ages 12 to 15) and the upper level is called Lyceum (High school) and lasts for three more years (ages 15 to 18). Lyceum studies can be General or Vocational.

Informatics was first introduced in the school curriculum during the school year 1984-1985 for Vocational schools, in 1993-1994 for Gymnasiums and in 1998-1999 for Lyceums.

Initially, teachers of mathematics and physics, who have attended seminars on Informatics or had a master degree in Computer Science, were appointed to teach the course of Informatics. Since 1992, graduates with a bachelor degree in Computer Science have started to teach Informatics at schools. As a sequence, nowadays, younger teachers have a bachelor degree in Computer Science but senior ones do not. Although teachers have tenure, every school year, teachers may be re-assigned to different schools, either by choice or for covering the needs of the educational system.

In the Interdisciplinary Unified Education Course

Framework (DEPPS) [20] and the new Curriculum (APS) were drawn up for compulsory education. School books were written based on the DEPPS and the APS guidelines and introduced to the school curriculum starting with the 2006-2007 school year.

Before September 2016 out of a total of 5,400 Elementary schools only 1,000 taught ICT (Information and Communications Technologies) for one hour a week in the first and second grade and for two hours a week in all other grades. Pupils were taught introductory programming courses in the last two grades. Since September 2016, ICT is taught only for an hour per week in all grades in every Primary school. In addition, the curriculum guidelines do not specify a particular programming environment and it's up to the teacher to choose one from those available. Therefore, students entering the Junior High school often come from different Elementary school backgrounds and do not have the same experience with programming.

In the Junior High school students are taught Informatics in all three grades too, but only for one hour a week. At the moment we performed our research, programming existed in the curriculum only in the third grade where fundamental algorithms and programming structures were formally introduced, using Logo-like environments.

In the first grade of the General Lyceum the Computer Science course is optional for the students (the curriculum includes programming). During the second grade all students are taught Informatics for an hour a week. The course includes the basic concepts of algorithms and programming structures using pseudo code syntax. The 3rd grade offers different curricular pathways. Students selecting the Economics and Informatics pathway have to take a two hour per week course that includes the development of computer programs using a special programming environment, similar to pseudo-code.

The first grade of Vocational Lyceum is similar to that of the General Lyceum, i.e. programming is optional. Starting with the second grade of Vocational Lyceum, students are offered several different pathways to select from, including an Informatics pathway. The Informatics pathway includes classes in Python and Java programming.

## III. RESEARCH QUESTIONS

Based on the theoretical background above, a questionnaire was devised and provided to teachers of Informatics in Greece having the following research questions in mind:

- 1) What type of programming environments have they used in class?
- 2) What was the dominant block- based programming tool?
- 3) What was the percentage of Junior High school students, having already been introduced to programming and what type of programming environment have they used.
- 4) Whether, according to their opinion, introducing students to block -based programming environments first, eases the transition to text-based programming.
- 5) Whether they had used both types of environments in teaching a programming concept (first using blocks and then writing commands) and if this technique

was successful.

- 6) Which they think is the appropriate age for students to transition from block-based programming to text-based programming.
- 7) Do their teaching experience, their employment status in different levels of education and their school location influence their answers?

## IV. MATERIALS AND METHODS

Quantitative research methods were used to explore the attitudes, values and beliefs of the research population [21]. The tool used to conduct the study was a structured and anonymous questionnaire addressing the research issues associated with this study. The construction of the questionnaire was driven by the intention of providing quantitative results that could be analyzed [22].

The questionnaire consisted of two parts. The first part included demographic questions and the second part questions concerning factors related to the type of programming environment (block vs. text).

During the investigation, the research team respected the anonymity and confidentiality of the respondents (Cohen et al., 2008).

Before distribution, the questionnaire was submitted to a content and face validity test. Experts agreed with the proposals incorporated in the questionnaire. In order to ensure its consequence and the stability of its results internal consistency reliability technique was applied. The Cronbach's alpha coefficient was 0.71 [23].

The survey's reference population was the teachers of Informatics in primary (1163 individuals) and secondary (5596 individuals) schools in Greece. The actual number of teachers was provided by the Ministry of Education, Research and Religious Affairs.

The questionnaire was distributed electronically to the personal emails of the research population (IT teachers of primary and secondary education in Greece, 6759 teachers in total). Weighting of the data was done using the variable "Teacher position" (primary or secondary) based on the actual ratio of teachers in primary (1163 or 17.2%) and secondary (5596 or 82.8%) education as provided by the Ministry of Education, Research and Religious Affairs. Hence, after weighting, the response sample consisted of 128 (or 17.2%) primary and 614 (or 82.2%) secondary school teachers.

The questionnaire, in Greek, was sent by email and the responses were received anonymously. Overall, 42 variables were coded. The statistical package SPSS (Statistical Package for Social Sciences V 20) was used to perform statistical analysis. The association between different variables was tested using the  $x^2$  independence test. For significance level a=0.05, a result was considered statistically significant when p-value < 0.05 [23]-[25].

# V. RESULTS

# A. Participants Profile

The sample consisted of 741 professionals. Their basic demographic characteristics are presented in Table I.

TABLE I: DEMOGRAPHIC OF PARTICIPANTS.				
	N	(%)		
Gender				
Male	424	57.2%		
Female	317	42.8%		
Teachers position				
Primary education	128	17.2%		
Secondary education	614	82.8%		
Status				
Permanent	712	96%		
Substitute	29	4%		
Years of teaching experience				
<20	544	73.4%		
>20	198	26.7%		
Current teaching position				
Elementary	134	18%		
Junior High school	250	33.7%		
General Lyceum	229	30.8%		
Vocational Lyceum	129	17.4%		
School location				
Capital District	348	46.9%		
Co-Capital District	108	14.6%		
other	285	38.5%		

As can be seen, the majority of participants were men (57.2%). As to their position, the majority (82.87%) was employed in secondary education and 17.2% in primary education. Almost all (96%) had a permanent position and only 4% were substitute teachers. More than two thirds (74.4%) had less than 20 years of experience and the rest 26.7% had more. We noted that during our research 18% of the participants taught in Elementary school, 33.7% taught in Junior High school, 30.8% taught at a General Lyceum and 17.4% taught at a Vocational Lyceum. Almost half of the teachers (46.9%), taught at a school in the Capital District, 14.6% of the participants' school was in Co-Capital District and 38.5% taught elsewhere in Greece.

## B. Data Description

Teachers were asked the type of programming environment they have used in class. In Elementary school, where no specific directions is set by the curriculum, the vast majority of them (83.7 %) chose to use block-based environments, 11.5% used both block and text-based programming environments and only 4.8% used only textbased programming. In Junior High school, where Logo-like environments are suggested by the curriculum guidelines, one third (34.4 %) of the teachers used text-based environments, one third (33.1%) used block-based environments and one third (32.5%) used both types. In General Lyceum 61.2% of teachers used both types of programming environments, 25.6% used text-based and 13.2% used only block-based environments. In Vocational Lyceum, half of the teachers (47.5%) used only text-based programming, an equal percentage (47.5%) used both programming types and only 4.9% used exclusively blockbased programming. These results are summarized in Table II.

TABLE II: TYPE OF PROGRAMM	ING ENVIRONMENT	USED IN CLASS.
Only block-based environments	Only text-based environments	Both types of environments

Elementary school	83.7%	4.8%	11.5%
Junior High school	33.1%	34.4%	32.5%
General Lyceum	13.2%	25.6%	61.2%
Vocational Lyceum	4.9%	47.5%	47.5%

Those who use block-based programming were asked to determine which specific environment they had used by choosing from a list containing "Scratch", "App Inventor", "Alice" and "other". In Elementary school, the vast majority (87.8%) of teachers that used block-based environments taught students programming using Scratch and the rest of them 12.2% used another programming environment, App Inventor and Alice excluded. In Junior High school, a similar percentage to that observed at the Elementary school, taught programming with Scratch (89.8%), whereas only 7.5% used another programming tool, 2.2% used App Inventor and 0.5% used Alice. In General Lyceum more than half of the teachers (56.1%) used App Inventor, suggested by the Ministry of Education, 29% used Scratch, 9.4% used another and 5.5% used Alice. In Vocational Lyceum, more than half of the teachers participating in the research (58.3%) used App Inventor in class, 26.4% used Scratch, only 4.2% used Alice and the rest of them (11.1%) used a different programming environment. We notice that in Elementary and Junior High school Scratch dominates, whereas at upper Secondary schools (Lyceums) App Inventor is the most popular among teachers. Table III summarizes these results. Teachers responding to the "other" option were given the unrestricted option to answer with the programming environment they used. We noticed that not one specific environment dominated their answers.

TABLE III: USE OF SPECIFIC PROGRAMMING BLOCK-BASED ENVIRONMENT.

	Scratch	App Inventor	Alice	other	
Elementary school	87.8%	-	-	12.2%	
Junior High school	89.8%	2.2%	0.5%	7.5%	
General Lyceum	29%	56.1%	5.5%	9.4%	
Vocational Lyceum	26.4%	58.3%	4.2%	11.1%	

Teachers teaching in Junior High schools were asked if their students had already been introduced to programming during their Elementary education. A significant percentage of 43% responded positively. Those who had responded positively were then asked to estimate the students' percentage that had already been introduced to programming. The majority of them (45%) responded that only 20% of their class students had been introduced to programming in the Elementary school, 31.8% responded that the class percentage was between 20% and 50% and only 22.6% of these teachers responded that more than 50% of their class had already been introduced in programming in the Elementary school (Fig. 1a). Teachers, whose students' percentage was more than 50%, were asked the type of programming environment their students had earlier used. The majority of them (78.4 %) responded that their students had been introduced to programming in Elementary school, via a block-based environment, 15.3% answered that their class students had been introduced both to block-based programming and text and only 6.3% of them answered that their students had been introduced to text programming exclusively (Fig. 1b).



Fig. 1. (a) Students prior knowledge in programming. (b) Type of prior programming experience.

Teachers were also asked to provide us their opinion about whether students' experience with block-based programming would help introduce students to text-based programming. The majority of them (71.2%) responded that teaching programming via a block-based environment first, facilitated the introduction to text-based programming, 22.1 % responded that it had no impact in the teaching process and only 6.7 % of the teachers thought that it was a negative factor (see Fig. 2a). We noted that over half of the participants (60%) have used both types of programming structure and the vast majority of them (81%) believed that this technique facilitated the educational process, 15% believed that it had no impact (see Fig. 2b).



Fig. 2. (a) Prior block programming knowledge and (b) effect of using both types of environments

Finally 80.3% of the teachers, who responded to our survey, believed that block-based programming must precede text-based programming in any case in the teaching process.

When teachers were asked to evaluate the most appropriate type of programming environment for Elementary school, the three Gymnasium grades, and the three grades of Lyceum, there seemed to be a "turning point" at the end of the lower Secondary school: 34.28% of the teachers, believed that students should transit from block-based programming to text programming during the 3rd grade of the Junior High school (Gymnasium), 23.19% believed that the transit should occur at the 1st grade of the Lyceum, 20.31% believed that text-based programming should be introduced during the last two Lyceum grades, 14.11% believed that the transit should occur at the 2nd Gymnasium grade, 5.458% believed that text-based programming should be introduced at 1st Gymnasium grade and only 2.645% believed that students should begin text programming in the Elementary school, without previous introduction to block-based programming (see Fig. 3).



Fig. 3. Proper grade for transition to text programming

#### C. Statistical tests

In order to determine whether teachers' opinions are associated to certain demographic characteristics, we examined if several correlations are worth mentioning.

The teacher's opinion regarding the impact of the students' previous experience with block-based programming on the introduction of text programming was strongly associated with the teacher's position- that is whether their position was in primary or secondary education-( $\chi^2$ =20.499,df=2, p-value=0.00), with the teacher's years of teaching experience, ( $\chi^2$  = 9.48, df =2, p-value=0.008),with their current teaching position during the research ( $\chi^2$  = 45.651, df =6, p-value=0.001) (see Table IV).

TABLE IV: CROSS TABULATION OF TEACHER DEMOGRAPHICS AND TEACHER OPINION ON THE IMPACT OF STUDENT' PREVIOUS EXPERIENCE WITH BLOCK-BASED PROGRAMMING TO STUDENT INTRODUCTION TO TEXT PROGRAMMING

	$\chi^2$	df	p-value
Teachers' position	20.449	2	0.00
Teaching experience	9.48	2	0.008
Current teaching position	45.651	6	0.00
School location	13.106	4	0.011

Similarly, the teacher opinion regarding which type of programming environment must precede the other in the teaching process is strongly associated with the teacher's position ( $\chi^2$ =24.081, df=1, p-value=0.00), with their experience( $\chi^2$ =9.786, df=1, p-value=0.02), with their current teaching position during the research ( $\chi^2$  = 31.507, df=3, p-value=0.00) and their school location ( $\chi^2$  = 9.864, df=2, p-value=0.007) (see Table V).

TABLE V: CROSS-TABULATION OF TEACHER DEMOGRAPHICS AND PERCEPTION REGARDING THE TYPE OF PROGRAMMING ENVIRONMENT MUST COME FIRST BLOCK VERSUS TEXT

COME FIRST, BLOCK VERSUS TEXT.			
	χ2	df	p-value
Teachers' position	24.081	1	0.00
Teaching experience	9.786	1	0.02
Current teaching position	31.507	3	0.00
School location	9.864	2	0.007

As expected, there was a significant correlation between, the teachers' perception of which type of programming environment must precede the other in the teaching process and which is the appropriate type of programming environment at Primary school ( $\chi^2$ =67.860, p-value=0.00) at 1st grade Gymnasium ( $\chi^2$ =98.694, p-value=0.00), at 2nd grade of Gymnasium ( $\chi^2$ =47.843, p-value=0.00) at 3rd grade of Gymnasium ( $\chi^2$ =21.712, p-value= 0.00) and at 1st grade of Lyceum ( $\chi^2$ =11.632, p-value=0.01) but we have no similar results for the last two grades of Lyceum.

We also noticed significant correlation, between the teachers' perception of the appropriate age of transition from block programming to text programming and their years of experience ( $\chi^2$ =19.730, df =5, p-value=0.01), the teachers' teaching position during the research ( $\chi^2$ =44.304, df=5, p-value=0.00) and their school location ( $\chi^2$ =24.790, df=5, p-value=0.006) (see TableVI).

TABLE VI: CROSS-TABULATION OF TEACHER DEMOGRAPHICS AND PERCEPTIONS OF THE APPROPRIATE AGE TO TRANSITION FROM BLOCK TO

TEXT PROGRAMMING.			
	χ2	df	p-value
Teaching experience	19.730	5	0.01
Current teaching position	44.304	5	0.00
School location	24.790	5	0.006

Finally, there is a strong correlation between the teachers having used both types of programming environments with their position ( $\chi^2 = 10.955$ , df=1, p-value=0.001) and their current school position ( $\chi^2=17.596$ , df=3, p-value=0.001).

## VI. DISCUSSION - CONCLUSIONS

Our first research question was to determine the type of programming environments teachers used in class in Greece. Although the curriculum mandated by the Ministry of Education provides no specific directions, teachers' answers were in accordance to the fact that block-based programming environments are used throughout the world as introductory tools [2] - [4].

Our second research question was to determine the dominant block- based programming tool used by teachers in Greek schools. Our research revealed that over 85% of the teachers that use a block-based environment in Elementary school and Junior High school use Scratch. These results were expected since Scratch is a dominant block-based introductory tool used mainly for primary education [18].

Our research also revealed that 43% of the Gymnasium students had already been introduced to programming and of these 78.4 % had used block-based environments. The low percentage 43% above can be explained by the fact that in the recent past, in Greece, ICT had been taught only in 20% of primary schools. In consequence, teachers in Gymnasiums had to teach students with varied or no background in programming. Thus, it is difficult for teachers to select the appropriate environment to use in classroom. The result of our research could help consultants on how to advise teachers in making the appropriate choice. We quote a participant response "the programming environment being chosen must be used throughout all Greek schools".

Note, that researchers in education have created teaching

environments that incorporate both block and text-based programming, such as Blockly and Pencil Code. We believe that such environments could help in teaching students with different programming background.

In accordance with previous findings by Armoni et al. [12] and Wolz et al.[13], this study indicates that the majority of teachers (71.2%) believe that introducing students to block-based programming environments first eases the transition to text-based programming later.

Furthermore, we found that more than half of the respondents (60%) had used both types of programming environments in the classroom during teaching a programming structure and the vast majority of them, 81%, believe that this technique is positive for the educational procedure. Such teaching techniques have been evaluated in the classroom by researchers [14], [15], [7] and their experimental results agree with our participants' point of view.

It is interesting that although the vast majority of teachers (80.3%) had the opinion that block-based programming must precede whatever the case in the teaching process, a relatively large percentage (19.7%) believe that students should first be introduced to text programming.

One of the primary tasks of this paper was to determine the appropriate age of students for transition from blockbased programming to text-based programming. In the UK, the curriculum suggests that such a transition should occur when students are 11 to 14 years old. Analysis of our results gave us a specific age for that transition. The teachers in our research sample believe that students in Elementary school and the two first grades of the Gymnasium should use a block-based environment and should be introduced to textbased programming at the 3rd grade of the lower Secondary school, when students are 14 years old.

School location, that is whether a teacher teaches in a metropolitan area or not, is associated with the participants' opinion on the impact of the students' previous experience to the introduction of text programming, with their perceptions of which type of programming environment must precede in the teaching process, as well as with their perception of the appropriate age of transition from block programming to text programming. This was anticipated since there is a geographical diversity in Greece among metropolitan areas and other regions. Seminars and other educational activities for teachers are more easily organized in metropolitan areas and is easier for teachers to meet and collaborate.

The above perceptions were also associated with the years of teaching experience. This was an important conclusion of our research. Experience is widely considered an important factor in many professions. In our case, younger teachers, have a degree in computer science. Although their teaching experience is limited, their theoretical knowledge is usually fresher.

Current teaching position was also associated to the above opinions as well as their initiative to use both types of programming environments during the teaching process. As mentioned before, the school curriculum in Elementary school gave no specific directions. The Gymnasium school curriculum suggests that Logo-like environments should be used. Finally, In General and Vocational Lyceum guidelines are more strict with emphasis on text-based programming. Thus teachers at lower class levels are more flexible in choosing the type of programming environments.

Currently, there is a tendency throughout the world on designing new curriculum in primary and secondary education in Informatics [18], [26], [27]. We believe that our results could be useful to stakeholders when designing new curriculum in Informatics, especially concerning the proposed type of programming environments per age.

## VII. FUTURE WORK

The authors plan is to further analyze the survey results, using log-linear statistical analysis techniques, involving more than two variables. The variables that will evaluate simultaneously are: (a) the teacher age, (b) the student age of transition to text programming (c) the school location metropolitan or not, etc. We expect this work to reveal estimated models for the variance in the observed frequencies outlined in this paper.

## ACKNOWLEDGMENTS

We would like to thank all our colleagues that participated in the survey for this research project and John Theodosiou and Iraklis Varlamis, who read this paper and made significant remarks.

#### REFERENCES

- [1] T. W. Price and T. Barnes, "Comparing textual and block interfaces in a novice programming environment," in Proceedings of the eleventh annual International Conference on International Computing Education Research, 2017, pp. 91–99.
- [2] J. Chetty, and G. Barlow-Jones, "Bridging the Gap: the Role of Mediated Transfer for Computer Programming," in International Proceedings of Computer Science & Information Technology, 2012, p. 43.
- [3] M. Kölling, N. C., Brown, and A.Altadmri, "Frame-based editing: Easing the transition from blocks to text-based programming," in Proceedings of the Workshop in Primary and Secondary Computing Education . ACM., 2015, pp. 29–38.
- [4] D. Weintrop, "Comparing Text-based, Blocks-based, and Hybrid Blocks/Text Programming Tools," in Proceedings of the eleventh annual International Conference on International Computing Education Research. ACM., 2015, pp. 283–284.
- [5] N. C. C. Brown, M. Kolling, and A. Altadmri, "Position paper: Lack of keyboard support cripples block-based programming," in 2015 IEEE Blocks and Beyond Workshop (Blocks and Beyond), 2015, pp. 59–61.
- [6] F. McKay and M. Kölling, "Predictive modelling for HCI problems in novice program editors," in Proceedings of the 27th International BCS Human Computer Interaction Conference. British Computer Society, 2013.
- [7] E. Kanidis, M. Karaliopoulou, and G. Menounou, "Students perceptions on software environments used for programming introduction," in Proceedings 8th Conference on Informatics in Education.CIE, 2016, pp. 303–313.
- [8] D. Parsons, and P. Haden, "Programming osmosis: Knowledge transfer from imperative to visual programming environments," in Proceedings of The Twentieth Annual NACCQ Conference, 2007, pp. 209–215.
- [9] D. Weintrop and U. Wilensky, "To block or not to block, that is the question: students' perceptions of blocks-based programming," in Proceedings of the 14th International Conference on Interaction Design and Children. ACM., 2015, pp. 199–208.
- [10] D. Weintrop and U. Wilensky, "Bringing Blocks-based Programming into High School Computer Science Classrooms," in Annual Meeting of the American Educational Research Association (AERA). Washington DC, USA., 2016.
- [11] B. DiSalvo, "Graphical qualities of educational technology: Using drag-and-drop and text-based programs for introductory computer science," IEEE Comput. Graph. Appl., vol. 34, no. 6, pp. 12–15, 2014.

- [12] M. Armoni, O. Meerbaum-Salant, and M. Ben-Ari, "From scratch to 'real' programming," ACM Trans. Comput. Educ., vol. 14, no. 4, p. 25, 2015.
- [13] U. Wolz, H. H. Leitner, D. J. Malan, and J. Maloney, "Starting with scratch in CS 1," ACM SIGCSE Bull., vol. 41, no. 1, pp. 2-3, 2009.
- [14] A. Wagner, J. Gray, J. Corley, and D. Wolber, "Using app inventor in a K-12 summer camp," in Proceeding of the 44th ACM technical symposium on Computer science education. ACM., pp. 621-626.
- [15] M. Dorling and D. White., "Scratch: A way to logo and python," in Proceedings of the 46th ACM Technical Symposium on Computer Science Education. ACM, 2015, pp. 191–196. [16] D. N. Perkins & G. Salomon, "Teaching for transfer. Educational
- leadership," Educ. Leadersh., vol. 46, no. 1, pp. 22-32, 1988.
- [17] Department for Education, "National Curriculum from September 2014 Retrieved December 15 2016 from https://www.gov.uk/government/collections/national-curriculum.," 2013.
- [18] N. C. Brown, S. Sentance, T. Crick, and S. Humphreys, "Restart: The resurgence of computer science in UK schools," ACM Trans. Comput. Educ., vol. 14, no. 2, p. 9, 2014.
- [19] M. Saeli, J. Perrenet, W. M. Jochems, and B. Zwaneveld et al, "Teaching programming in secondary school: a pedagogical content knowledge perspective," Informatics Educ., vol. 10, no. 1, 2011.
- [20] DEPPS APS Informatics, "Interdisciplinary Unified Education Course Framework and the new Curricula. Educational Institute. Retrieved December 15, 2016 from http://www.pischools.gr/programs/depps/," 2003.
- [21] C. Robson and K. McCartan, Real world research: a resource for users of social research methods in applied settings.
- [22] L. Cohen, L. Manion, and K. Morrison, "Research methods in education," in Routledge., 2013.
- [23] I. Apostolakis, & M. A. Stamouli, "Validity and reliability assessment of quantitative research questionnaires in health units: The case of a questionnaire concerning the evaluation of a nursing services management information system of a hospital.," Stat. Rev., vol. 2, no. 1, pp. 3-25, 2006.
- [24] I. Apostolakis, A. Kastania, & C. Pierakou, "Statistical Data Processing in Health Sector," 2003.
- [25] I. Apostolakis and M. Stamouli, Askiseis ipologistikis statistikis stin igeia (teyxos A). Papazisis Pub., 2007.
- [26] S. Iyer, F. Khan, S. Murthy, V. Chitta, M. Baru, and U. Vishwanathan, "CMC: A Model Computer Science Curriculum for K-12 Schools.," 2013.
- [27] K-12 Computer Science Framework Steering Committee, "K-12 Computer Science Framework. Technical Report.ACM, New York, NY, USA.," 2017. [Online]. Available: https://k12cs.org/wpcontent/uploads/2016/09/K-12-Computer-Science-Framework.pdf.



Margarita Karaliopoulou was born in Athens. She obtained a degree in Mathematics from University of Patras in 1991, a MSc in Informatics and Operational Research in 1993, a MSc in Statistics and Operational Research in 2004 and a PhD from University of Athens. with a dissertation entitled " Discrete Semi-Markov Procedures and applications in word Occurrences".

She has worked for several years as a teacher of

Informatics in secondary education. Since November 2018, she has been a EDIP member in the Department of Mathematics at the National and Kapodistrian University of Athens. Author of a book and several papers among which the paper entitled "On Discrete Time Semi-Markov Chains and Applications in Words Occurrences" Communications in Statistics-Theory and Methods. 37(8) 1306-1322 (Chryssaphinou O., Karaliopoulou M and Limnios N.)

Dr Karaliopoulou is a member of the Hellenic Mathematical Society and of PAPEDE (Panhellenic Association of Secondary education) cofounder of PEKAP (Panhellenic Association of teachers in Informatics).



Ioannis Apostolakis was born in Chania, Crete. He holds a degree in Mathematics (1983), MSc in Informatics. Operational Research and Educationissues (1987), and a PhD in Health Informatics (1993) at the University of Athens, Greece.

He works as Director in one of the Vocational Institutes at the Ministry of Education, Research and Religion Affairs in Athens. He has been a scientific researcher for several years at the Department of

Clinical Therapeutics of the University of Athens. He has research and educational activities on Health Informatics and Education in several universities (1995-2018). He has taught a wide range of courses at undergraduate and postgraduate level: Health Informatics, Information Systems, Computational Statistics I, II, e-Government, e-Democracy, Management of Virtual Communities, Research Methodology and the Internet, Teaching of Computer Science, Data analysis using statistical techniques. He has worked as a Computer Specialistat the Ministry of Interior (1989-1995) and at the National School of Public Administration as Head of Informatics Program (1995-2005).

Dr. Apostolakis has also been a program committee member of many International Conferences and as a reviewer for several journals in the area of e-learning and Healthcare service management.



Evangelos Kanidis was born in North Greece. He studied Mathematics at the University of Athens. He received the MSc in Informatics from the Informatics Department of Athens University and the PhD in Cognitive Science from the Interdisciplinary Graduate Program of Athens University in "Basic and Applied Cognitive Science", in 2007.

He worked as a teacher in secondary education for several years and as a scientific researcher and tutor in Piraeus University of Applied Science. He is

currently an educational consultant in Informatics at the Ministry of Education. His research focuses mainly on the educational aspects of informatics and on ways to improve classroom IT teaching in primary and secondary education. He authored two books, co-authored nine books and published more than 40 articles in various scientific journals and international conference proceedings.