TEACHING ALGORITHMS PROFILE-ORIENTED: A PROPOSED METHODOLOGY TO ELEMENTARY SCHOOL

Fernando Moreira, Maria João Ferreira

Universidade Portucalense (PORTUGAL)

Abstract

One of the first challenges in the programming field, apart from ones related to the traditional teaching methods, is the introduction of technological tools that can help improving children's learning experience by presenting them with the most appropriate problems according to their ages. This choice is of great importance since it allows students to develop their capacity of abstraction in a natural way. Moreover, it also improves students' creativity and problem-solving abilities. In order to overcome these challenges, we propose a methodology based on VARK questionnaire that defines the students' profile and, consequently, what are the most appropriate educational materials based on technologies. Once the students' profiles are defined it is suggested the use of Scratch programming language to develop the student's skills on the resolution of problems through the following concepts: sequencing, repetition and conditions.

Keywords: programming; VARK questionnaire; methodology; instructional materials; programming tools.

1 INTRODUCTION

Technology is still undergoing a rapid paced growth; over time, people are increasingly involved in technology issues as regards various aspects of their daily lives, although subjects such as education can be emphasised. As a matter of fact, people of almost all ages use technology in order to, directly or indirectly, support their learning. Currently, technologies are being taught in several schools, including primary schools [1]. In this context, digital qualification is an increasingly necessity that has an impact both at work and in everyday life. However, the number of people involved in the use of technology in education is still very low. As the European Union expected, 90% of the European jobs require ICT skills [2], and 900,000 jobs in the ICT areas, or about 2 million if the STEM (Science, Technology, Engineering and Mathematics) is included, will still be unfilled, by 2020. Even though, less than 15% of European students have access to ICT education in school.

The acquisition of the skills required for developing computer programs is one of the first challenges that students have to cope with when starting a degree in Computer Science or related field of study. Therefore, the course unit Algorithms and Programming is the one that holds a higher level of difficulty, due to the need to develop the abstraction ability, which is unknown or little practiced by many of the students [3].

Barcelos [4] emphasizes in his study the high students' failure and dropout rates in curriculum unit Algorithms and Programming, based on a research conducted during the school years 2008 and 2009 (2nd half of 2008 and the 1st and 2nd half of 2009). In this study, it was assessed the students' failure percentages, which exceed 50% in all semesters.

In order to increase the technological literacy among the society, some countries are training elementary school teachers for teaching programming. The reason for this commitment is related to the fact that younger children reveal a natural tendency to learn programming languages more easily than adults, as it happens, for instance, with foreign languages. For example, Fessakis et al. [5] present a case study on the dimensions of problem solving using computer programming for children 5-6 years old. The results of this study show that children who attended the new learning activities developed mathematical concepts, problem solving and social skills.

The proper use of computers, in addition to traditional teaching methods, can help improve children's learning experience by presenting them with problems in a most appropriate way, concerning their ages. The idea is to incorporate learning into something that children perceive as something funny and enjoyable. Thus, problems such lack of interest or misunderstanding a certain issue should be easily overcame [6].

ISBN: 978-84-606-8243-1

Razak et al. [7] investigated how children see technology according to their perspectives concluded that children really want a technology that is ubiquitous, wearable, interaction and self-centred. In the same way McKenney & Voogt [8] presented an interesting study which analysed the access of 167 young children to technologies, their perceptions and use of technology inside and outside the school environment. The results obtained point the fact that children's attitudes toward computers are positive, and also highlight the importance of exposing children to computers at home and in educational settings.

The US experience [9] indicates that children under the age of 7 years should be started in the programming area in a funny and acceptable way so they can think about procedures and algorithms.

In Portugal, the Ministry of Education and Science (MES) [10], will *"... launch the pilot project "Introduction to Programming in the 1st cycle of basic education", challenging the public schools of Portugal to take part in it in the next school year (2015/16) with its students of 3rd and 4th years of schooling.*" In this context, in the next school year, 34,000 children will be enrolled on programming lectures in Portugal through an initiative that covers a total of 240 groups and 1805 classes. The initiative also includes the training of 720 teachers.

The main aim of the current research is to identify and to choose the most appropriate tool for the introduction of basic programming concepts to elementary school children. This proposal will be part of the third component of the Profile-oriented algorithms teaching methodology (PoATM) [3]. There are two main reasons for selecting this type of approach. First, recent studies have shown that children are able to learn while they are playing [11], either consciously or unconsciously. Second, according to cognitive development Piaget theory [12], children are entering the concrete operational state of cognitive development. The concrete operational state is the third of four states concerning cognitive development and is characterized by the enhancement of logical thinking and problem-solving skills that are both intended to be developed once this proposal is applied.

This article will be further organized as follows. In the next section, a background of the addressed subjects is presented. The state of the art is described in section III. In sections IV and V it is presented the methodology and the respective proposed tool. Finally, in the last section, conclusions are discussed.

2 BACKGROUND

2.1 Learning programming

The definition of programming provided by Merriam-Webster dictionary (http://www.merriam-webster.com/dictionary/programming) is "*the act or job of creating computer programs* ". This apparent simplicity is contradicted by the highest dropout rates in universities programming courses, as mentioned in Barcelos [4]. Similarly, Tan, et al. [13] showed that the high dropout rate and failure rate in programming subjects have drawn serious attention within researchers to investigate their causes and come out with possible solutions. The main reason of such negative results in programming courses is because most of the undergraduates did not have any programming experience before. When the students arrive to the university they start learning programming in a single context, before understanding its structure and style. The same authors [13] argue that "*this might lead to a negative programming habit which may affect the flexibility of learning another programming language in different context*" and, consequently, students "*think that programming is difficult to learn and use. Therefore, undergraduates might refuse to learn programming skills unconsciously.*"

Jenkins [14] showed that "*few students find learning to program easy*" and "*programming is a complicated business*" because is not need a single skill, but students need to have a set of skills, such as "*problem solving ability and some idea of the mathematics underlying the process are essential*". Programming is a new subject for many students who take programming courses. Dijkstra [15] argues that learning is a slow and gradual process of transforming the "*novel into the familiar*". A particular feature of programming is that it is "*problem-solving intensive*" [16] requires a considerable amount of effort for often a very modest return.

Rahmat, et al. [17] argued that "*the programming subject itself is a main reason that makes the courses difficult because it is new to most of the students*.", and because "*programming is a hierarchical skill where students acquire a basic skill before advance level*." In this study several major problems regarding reference materials, lectures and lab session approaches, problem solving ability, time management and self-confidence were identified.

2.2 Learning Styles

Throughout the past few years, several researches have been conducted as regards the students' preference concerning the type of teaching / learning, since they have different preferable ways of learning as well as self-abilities [18]. However, it is also worth noting that a student learning preference may also vary throughout time, along with his maturation, and acquisition of different skills. All the various models developed to explain the different learning styles are here grouped into four general categories: a) personality models, b) information processing models, c) models of social interaction, d) models of instructional preferences [19]. The incompatibility between the preferred learning style and educational strategy used by each teacher leads to several consequences, as presented in [20]: (i) inattentive students; (ii) discouragement by the course; (iii) low grades; (iv) lack of interest; (v) hostilities and (vi) limited cooperation.

When analysing in particular the category d) – models of instructional preferences – the model which has gained more popularity, is the Visual, Aural, Read/write, and Kinesthetic questionnaire (VARK) [21]. VARK questionnaire (http://vark-learn.com/the-vark-questionnaire) is simple and has benn specifically designed to, on the one hand, help students meeting their learning profile more effectively and, on the other hand, help teachers becoming more sensitive as regards the variety of learning strategies needed to reach all students [22]. However, this questionnaire cannot be used to draw conclusions about the strengths or weaknesses of an individual during the learning process [23]. The VARK questionnaire consists of sixteen multiple-choice questions with four items each, corresponding to the four modalities V, A, R and K [21]. This tool is useful for teachers who are seeking to develop additional teaching and learning strategies [24], which should be more appropriate to the student's profile.

A student who has a V-type profile prefers to learn using graphics, symbolic representations, or video materials, or through the auditory input such as teacher's explanations or presentations. Obtaining an A type profile often indicates that a student perceives the information better if it is auditory and verbal. This type of students prefers discussions, explanations or lectures, i.e., a more traditional behavioural approach (expositive classes). The student belonging to the type R profile prefer to get the input and to produce output through reading and writing, which represents information in the form of text instead of sound or images. Students with this profile are more favourable for self-study [25], [26]. The kinesthetic (K) profile fancies more practical experience rather than other modes of perception. For a K-type profile student it is worth highlighting the importance of real or simulated practice; students can achieve better learning outcomes from practical demonstrations of teaching topics. Nevertheless, it should be mentioned that no teacher or student is restricted to only a learning preference mode.

3 STATE OF THE ART

So far, to the best of our knowledge, there is no work that uses VARK questionnaire for profiling, further development of materials and to choose what are the tools that should be used by students to learn programming. Conversely, the literature discussing teaching programming in early ages is quite extensive, as discussed below.

Krajnović, et al. [9] showed the advantages of replacing procedural programming languages (Pascal, Basic and Logo) and traditional forms of education with object oriented programming languages and to modernize different approaches towards IT education. However, in this work students' profiles were not considered.

Pardamean, et al. [27] introduced the idea that Logo programming language improves students' creativity and problem-solving skill among the best graded students. In the same way, the authors did not considered students' profiles in the assumptions of the programming language choice.

In [28] the authors argued that Scratch is a visual programming environment that is widely used by young people. The results discussed in this paper showed that students could successfully learn important concepts of computer science. However, the choice for Scratch was carried out without taking into consideration the student's profile.

In [29] the authors discussed an approach to teach programming that would allow elementary school children to adopt basic problem solving concepts. This approach consists of four different phases. In the first phase the authors conducted a verbal interview with the children to familiarize themselves with some basic concepts. In the second phase children are introduced with the system rules. The third phase consists of displaying the robot in simulated environment. Lastly, the fourth phase consists of executing the program developed in previous phase. Similarly to the previous studies, these authors did not consider the student's profile.

Alepis and Virvou [30] presented a mobile learning platform that targets elementary school students in order to teach them basic programming principles, named m-AFOL. The results obtained were very interesting: they concluded that the mobile facilities of the resulting intelligent tutoring system are highly acceptable from young learners who enjoy using mobile devices while learning. Similarly, the authors [31] showed that tangible programming tools enabled children to easily learn the programming process, which was previously considered as difficult. The authors developed a tool to improve the programming language experience by considering and utilizing characteristics of procedural language: elements such as sequence, repetition, condition, function, and parameter. However, as mentioned previously, the authors of both works [30] and [31] designed the platforms without considering students' profiles.

4 PoATM: SELECT TOOLS AND DEVICES

In general thesis, we can state that it is important that teachers consider the learning styles of students when designing and developing programming teaching materials. Although it may not be essential for teachers to test and instruct students in all situations, testing them with the VARK instrument – simple and fast – can provide important information that is beneficial to create an effective learning environment. Being aware of different learning styles and considering its impact on learning environments are the two first steps towards an essential understanding of the students' profiles when developing activities to teach programming.

4.1 Research methodology

The obtained results are be based on a research technique that makes use of pre and post-tests that measure the knowledge acquired by students in a learning activity. Such tests consist on the application of a set of questions that must be answered by students before and after a particular activity. In particular, the pre-test allows assessing the participants' level of knowledge on an issue that will be approached before training. The post-test consists in applying a set of questions with the same level of difficulty, in order to assess the evolution of the acquired skills [32].

The comparison between the pre-test and post-test notes, together with the profile obtained through the VARK questionnaire, allows the understanding of the benefits of the activity and determining what are the appropriate materials to enhance increasing the knowledge of the students and, consequently, decreasing failure and dropout rates. Furthermore, it should also contribute to the identification of the gaps present in the teaching-learning process regarding students that hold no predominant profile.

4.2 Methodology

In order to counter the problem addressed in section 2.1 we proposed a methodology [3] that should help elementary school children acquiring basic programming concepts. As discussed previously it is important to define the students profile in order to help designing and developing teaching materials. This assumption was the basis for the development of the methodology that is here proposed (Fig. 1). It consists of five different steps with a defined sequence: (A) Analysis of student profile; (B) Construction didactic material; (C) Select tools and devices; (D) Defining evaluation methodologies and (E) Monitoring and control.

Teachers should consider the different learning styles of students while designing the teaching programme and throughout the development programming activities. Despite the fact that it may not be highly essential for teachers to test and instruct students in all situations, testing them with the VARK instrument - simple and fast - can provide important information to create an effective learning environment. It should be once again highlighted that being aware of learning styles and considering its impact on the learning environments are the two first steps towards an essential understanding of the profile of the students when developing programming skills.

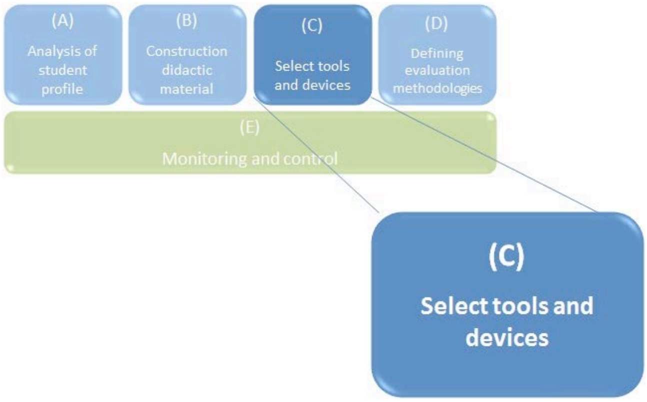

Figure 1. Methodology steps.

In the component (A) the VARK questionnaire is used to analyse the profile of each one of the students who attend programming courses in order to understand what kind of teaching materials need to be built and made available for (B) (for instance, it is assessed if those materials should be more descriptive or visual, etc). Based on the results and constructed materials it is necessary to carry on their distribution to different types of devices, namely, if the model is applied to classroom teaching, distance learning (e-learning), blended learning (b-learning), or to be used only in mobile devices (mobile learning, m-learning) (C). For instance, the use of e-learning or m-learning demands the production of materials appropriate for such type of devices. Some of the most important components in the teaching-learning models are the used evaluation methodologies (D). The whole methodology has a monitoring and control process (E) which allows the adjustment of the materials, devices used and evaluation methodology, as long as students and teachers understand these needs.

4.3 Select tools and devices

One of the great issues when choosing a tool to teach programming is the large amount and variety of tools, which makes it a great challenge. As there are several tools the following questions must be answered before the selection: Where to start? and How to choose the best language / programming environment that meets the needs of primary school students?

Due to the large number and variety of languages / programming environments it is necessary to establish a set of characteristics: (i) Number of supported languages; (ii) What are the operating systems that support (Windows, Mac, Linux, Other); (iii) Types of programming languages (Image or Picture Languages, Block or Bubble Languages, Programming Languages); (iv) Difficulty level (Initial, Intermediate, Advanced).

The features concerning the type of languages (iii) are:

1. Image or Picture Languages (IoPL) — these languages teach computer science and programming concepts through the manipulation of images.
2. Block or Bubble Languages (BoBL) — these languages use blocks and sometimes actual code to perform tasks, for example, spinning an image or making an image by following a specified route. These languages bridge the gap between picture languages and traditional languages.
3. Programming Languages (PL) — these languages are what professionals think of as programming languages. Programming languages can be learned step by step as people learn how to perform tasks and create applications (C, C #, Java, Logo, etc.).

In Table 1 there is presented one set of 20 tools which demonstrates the stated quantity and diversity of programming tools. It should be mentioned that, due to the target audience, it was not considered any language of the third type (iii.3).

Table 1. Surveyed tools

Tool	N. of languages supported	Operating System				Types of programming languages		Difficulty level		
		W	M	L	O	IoPL	BoBL	B	I	A
Gamestar Mechanic	1	X	X					X		
GameMaker	15	X	X	X	X			X		
Scratch	50	X	X	X			X	X		
Storytelling Alice	1	X						X		
Alice	1	X	X	X					X	
Kodu	8	X			X		X		X	
StarLogo TNG	1	X	X	X					X	
Pygame	1	X	X	X	X					X
Stencyl		X	X	X	X			X		
Turtle Art	8	X	X	X				X		
kidsruby	3	X	X	X	X				X	
Code Kingdoms	1							X		
Tynker	1				X		X	X		
SiMPLE	1	X						X		
Move the Turtle	1		X			X				
Hopscotch	1		X					X		
Daisy the Dinosaur	1		X					X		
Cargo-Bot	1		X				x	X		
RoboMind	24	X	X	X			X	X	X	
NXT-G	1	X	X	X			X	X	X	X

W – Windows; M – Mac; L – Linux; O – Other

IoPL – Image or Picture Languages; BoBL – Block or Bubble Languages

B – Beginner; I – Intermediate; A – Advanced

As a final remark, it was concluded that Scratch is the most appropriate tool for elementary school Portuguese students (targeted audience of the present study). This is the outcome of an analysis carried out with the four characteristics defined above ((i) Number of supported languages; (ii) What are the operating systems that support; (iii) Types of programming languages; (iv) Difficulty level), and taking into account that the target audience have no training in any foreign language (Scratch is available in a Portuguese version, in contrast to the majority of the tools). It is interesting to note that this result goes against what was proposed in [10].

Scratch was chosen by the MES since it is a programming language that has emerged as the most popular for young people all around the world who are learning how to code [33]. The National Science Foundation has supported the initial development of Scratch [34], the study of collaboration in the Scratch online community [35], the design of resources to support educators working with Scratch [36], and the development of a cloud-based version of Scratch [37]. This language is freely available and has been translated into more than 50 languages, which include Portuguese language. More than 4 million projects have been shared in the Scratch website, with over 6,000 new projects shared every day. The Scratch online community harbours more than 21 million comments and hundreds of millions of other recorded interactions and it continues to grow rapidly (http://scratch.mit.edu/statistics/).

As a complement to the choice of Scratch, it is verified that the MES will provide training to use this tool efficiently. It will take place in an e-learning system during the coming months of June and July. Schools will also have access to a monitoring system and online support (http://programacao1ceb.dge.mec.pt) through a practical community hosted on the initiative website.

The selection of devices where the Scratch will be used in the teaching-learning process in primary schools is also very important. Thus, as presented at Scrach Wiki "*Scratch is only available as a desktop application, meaning it runs on laptops and desktop computers. Scratch is currently not available as a mobile device app. However, the Scratch Team is developing an app for tablets that will run on the larger mobile devices. An HTML5 project viewer that will allow projects to run on any devices is also in development from the Scratch community*" [38]. In this context, the devices which will run the Scratch, for the programming of teaching in primary education, will be only computers and laptops.

5 CONCLUSIONS

To increase the technological literacy in the society, some countries are currently including programming lessons in teaching-learning programs at all educational levels, but mainly in primary school. The reason for this commitment regards the fact that younger children reveal a natural tendency to learn foreign languages more easily, which should also happen with programming languages. Such natural tendency will enhance the acquisition of skills to develop children's abstraction capability and improves students' creativity and problem-solving abilities.

In this context, the main goal of the current research was the proposal of a methodology based on VARK questionnaire to define the profile of students and, consequently, what are the most appropriate materials to teach programming according to the profile of students, when using Scratch as a programming language.

It is still intended, as mentioned, in an ongoing research, to test the proposed methodology and allow further development based on the results obtained from the methodology application in the next school year in schools that belong to the MES project.

REFERENCES

[1] Virvou, M., Alepis, E., Mpalasis, K. (2013). Evaluation of a multimedia educational tool for geography in elementary schools. Proceedings of International Conference on Information Communication Technologies in Education, ICICTE, pp. 364-374

[2] Tech (2015). Should computer science be taught in elementary schools in Europe? http://www.debatingeurope.eu/2015/01/22/should-computer-science-be-taught-in-elementary-schools-in-europe/#.VTofGKbw9bo

[3] Moreira, F. and Ferreira, M.J. (2015). Profile-oriented algorithms teaching: a proposed methodology. CISTI 2015. Accepted for publication.

[4] Barcelos, R. (2012). O Processo de Construção do Conhecimento de Algoritmos com o Uso de Dispositivos Móveis Considerando Estilos Preferenciais de Aprendizagem. Tese de Doutorado. UFRGS.

[5] Fessakis, G., Gouli, E. and Mavroudi, E. (2013). Problem solving by 5-6 years old kindergarten children in a computer programming environment: A case study. Computers and Education Volume 63, pp. 87-97

[6] Meluso, A., Zheng, M., Spires, H. A. and Lester, J. (2012). Enhancing 5th graders' science content knowledge and self-efficacy through game-based learning. Computers & Education, 59(2), pp. 497-504.

[7] Razak, A., Salleh, K. and Azmi, N.H. (2013). Children's technology: How do children want it? Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) Volume 8237 LNCS, pp. 275-284

[8] McKenney, S. and Voogt, J. (2010). Technology and young children: How 4-7 year olds perceive their own use of computers. Computers in Human Behavior Volume 26, Issue 4, pp. 656-664

[9] Krajnović, I., Bakić-Tomić, L. & Markovac, V. (2013) Object Oriented Programming in primary education.

[10] Minitério da Educação e Ciência. (2015). Projeto “Iniciação à Programação no 1.º Ciclo do Ensino Básico”. http://erte.dge.mec.pt/index.php?action=view&id=1538&date_id=1614&module=calendarmodule&section=9

[11] Prensky, M. (2008). Students as designers and creators of educational computer games: Who else? British Journal of Educational Technology, 39(6), pp.1004-1019

[12] Wadsworth, B. and Gray, W. (2004). Piaget’s theory of cognitive and affective development. Pearson/A and B.

[13] Tan P.-H., Ting, C.-Y. and Ling, S.-W. (2009). Learning Difficulties in Programming Courses: Undergraduates’ Perspective and Perception. International Conference on Computer Technology and Development. pp. 42-46 DOI: 10.1109/ICCTD.2009.188

[14] Jenkins, T. (2002) On the Difficulty of Learning to Program. http://www.psy.gla.ac.uk/~steve/localed/jenkins.html

[15] Dijkstra, E. (1989). On the Cruelty of Really Teaching Computing Science. Comm. ACM, Vol.32, pp 1398-1404.

[16] Perkins, D., Schwartz, S. and Simmons, R. (1988) Instructional Strategies for the Problems of Novice Programmers. In R. E. Mayer (ed), Teaching and Learning Computer Programming, pp 153-178.

[17] Rahmat, M., Shahrani, S., Latih, R., Yatim, N., Zainal, N. and Rahman, R. (2012). Major Problems in Basic Programming that Influence Student Performance. Social and Behavioral Sciences. Volume 59, 17 October 2012, pp 287–296. doi:10.1016/j.sbspro.2012.09.277

[18] Kumar, A., Smriti, A., Pratap, A. and Krishnee, G. (2012). An analysis of gender differences in learning style preferences among medical students. Indian Journal of Forensic Medicine and Pathology, 5 (1), pp. 9-16.

[19] Claxton, S. and Murrell, P. (1987). Learning styles: implications for improving education practices. ASHE-ERIC higher education report 4. Washington, DC: Association for the Study of Higher Education.

[20] Lindermann, V., Tarouco, M. and Bercht, R. (2008). Estilos de Aprendizagem: um estudo de casos em turmas de algoritmos e programação. XIX Simpósio Brasileiro Informática e Educação – Fortaleza – CE,.

[21] Bernardes, E. and Hanna, M. (2009). How do management students prefer to learn? Why should we care? International Journal for the Scholarship of Teaching and Learning, 3 (1), pp. 1-12.

[22] MERLOT (Multimedia Educational Resource for Learning and Online Teaching): the active learning site with VARK learning styles inventory. Available at www.merlot.org/merlot/viewCompositeReview.htm?id=145206

[23] Bednarik, R. and Fr¨anti, P. (2004). Survival of students with different learning preferences. The 4th Annual Finnish / Baltic Sea Conference on Computer Science Education, Kolin Kolistelut/Koli Calling, pp. 121–125.

[24] Ramayah, M., Sivanandan, P. and Nasrijal, N. (2009). Preferred learning style: Gender influence on preferred learning style among business students. Journal of US-China Public Administration, 6 (4), pp. 65 78.

[25] Fleming, N. (1995). I'm different; not dumb. Modes of presentation (VARK) in the tertiary classroom, in Zelmer,A., (ed.) Research and Development in Higher Education, Proceedings of the Annual Conference of the Higher Education and Research Development Society of Australasia (HERDSA), Volume 18, pp. 308–313. http://www.itu.dk/people/metteott/ITU_stud/Speciale/L%E6ring/learningStyles/different_not_dumb.pdf.

[26] Fleming, N. (2001). Teaching and Learning Styles: VARK Strategies. Honolulu Community College. http://www.vark-learn.com/english/index.asp.

[27] Pardamean, B. and Honni, E. (2011). The Effect of Logo Programming Language for Creativity and Problem Solving.

[28] Meerbaum-Salant, O., Armoni, M. and Ben-Ari, M. (2013). Learning computer science concepts with Scratch, Computer Science Education, 23:3, 239-264, DOI: 10.1080/08993408.2013.832022

[29] Zaharija, G., Mladenović, S. and Boljat, I. (2013). Introducing programming concepts to elementary school children. 4th International Conference on New Horizons in Education. (10). pp. 1576–1584 doi:10.1016/j.sbspro.2013.12.178

[30] Alepis, E. and Virvou, M. (2015). Evaluation of a mobile programming learning platform for elementary school students. International Journal of Information Technology and Computer Science (IJITCS). 15(2).

[31] Kwon, D., Kim, H., Shim, J. and Lee, W. (2012) Algorithmic Bricks: A Tangible Robot Programming Tool for Elementary School Students IEEE Transactions on Education, 55(4), pp. 474-479

[32] I-TECH. (2008). Technical Implementation Guide. Guidelines for Pre- and Post-Testing. University of Washington. Seattle, Washington. USA.

[33] Resnick, M., Silverman, B., Kafai, Y., Maloney, J., Monroy-Hernández, A., Rusk, N., Silver, J. (2009). Scratch: Programming for all. Communications of the ACM, 52(11), 60. doi:10.1145/1592761.1592779

[34] Maloney, J., Resnick, M., Rusk, N., Silverman, B., and Eastmond, E. (2010). The Scratch Programming Language and Environment. Trans. Comput. Educ., 10(4), 16:1–16:15. doi:10.1145/1868358.1868363

[35] Roque, R., Rusk, N., and Blanton, A. (2013). Youth roles and leadership in an online creative community. In Computer Supported Collaborative Learning Conference Proceedings, Volume 1, International Society of the Learning Sciences.

[36] Brennan, K., and Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. Proceedings of the American Educational Research Association (AERA) annual conference.

[37] Dasgupta, S., and Resnick, M. (2014). Engaging Novices in Programming, Experimenting, and Learning with Data. ACM Inroads, vol. 5, no. 4, pp. 72-75. DOI: 10.1145/2684721.2684737

[38] Scratch. (2014). Scratch on Tablets. http://wiki.scratch.mit.edu/wiki/Scratch_on_Tablets