Instructional Media and Teaching Methods for Engaging Children with Computer Programming

Conference Paper · July 2014

DOI: 10.1109/ICALT.2014.225

1 author:

Varvara Garneli
Ionian University
13 PUBLICATIONS **122** CITATIONS

SEE PROFILE

Instructional Media and Teaching Methods for Engaging Children with Computer Programming

Varvara Garneli
Ionian University, Department of Informatics
Corfu, Greece
C13garn@ionio.gr

***Abstract*— Technology - based learning could help students achieve fundamental abilities and skills like computational thinking and creativity. Engaging young students in computer science concepts and programming with a creative and enjoyable way is a challenging issue. We are aiming to apply and evaluate constructionist or apprenticeship techniques and visual programming tools that could assist and improve the learning activity.**

Keywords- CS; programming; visual programming environments; constructionism; programming pedagogy

I. INTRODUCTION

The main goal of this Ph.D. project is to explore instructional media and teaching methods for learning computer programming in an engaging way. In order to achieve this goal, this project will empirically understand the current teaching practices and technologies. In particular why and how students and teachers use the current programming tools in the ways that they do; and how these tools can be redesigned in order to optimize the teaching of programming.

II. MOTIVATION

Technology Enhanced Learning (TEL) could help students to develop higher – order skills like critical thinking, analysis, or scientific inquiry. There are several technology applications which support active engagement, team work, satisfied degree of human computer interaction and "real world" context. Computer technology's implementation is an interesting issue that needs further research [15]. Currently, there are multiple efforts to broaden participation in Computer Science (CS) and introduce computational literacy to young students [18], [23]. Education in schools should also be raising young students' interest in IT, specifically in CS subjects. Another crucial aspect is the low number of female students in CS subjects. Females more often than males choose disciplines like linguistics, cultural studies, and arts. In school, girls typically show less interest in CS topics; something that later prevents them from studying and pursuing a CS career [20]. Many teachers support that there is a lack of interest among young students about computer science. This could be explained from the curriculum content in the primary and the secondary education. Students find boring to focus in skills like typing, working with spreadsheets etc. Additionally, factors like cost, real life relevance or the existing stereotypes among students, parents and teachers must also be considered [4][5]. Nevertheless educators should help their students to achieve creative learning instead of only using the technology tools. Already, countries like the U.K. are planning to set up a new computer science curriculum for Primary and Secondary Education [25]. Under this perspective, engaging young students with computer science concepts and particular programming in a typical school environment seems a challenging issue that needs more research.

The early programming languages were too difficult mostly because of their syntax. Also the supported activities were not interesting or they were too complex for children. Recent programming languages like Scratch, Alice or Greenfoot offer a usable environment with many opportunities. Deciding which one will be used might depend on the age, the gender, the activity's type or goal etc [17], [21]. Additionally programmable hardware platforms could engage children in creative activities like robots [6]. Recently wearable computing was developed. It is an approach which integrates electronics and computers in materials like fabric [8]. These types of technology which connect coding ability with physical kits could make programming more excited. Another proposed approach is the mobile environment use. It is a tangible technology, very common and famous among young people. Educators could take advantage of devices like tablets and smart phones for motivating in computer science topics [22]. Finally, some researchers additionally use unplugged computer science activities [7] or even textual programming languages [12]. Therefore, it would be interesting to study and evaluate which characteristics of programming environments and how can motivate young students and also make learning effective.

Studying the pedagogy of programming is also an important issue. It is not enough to know the tools and their characteristics. It is necessary to understand, in a theoretical and practical way, how children come to understand programming and which the best way to teach is [10]. Constructionists provide students the opportunity to create their own projects. Applying this idea in a programming environment, students can create interesting activities or even their own video games. Additionally, teachers can take advantage of the students' enthusiasm for games in their effort for better learning [13]. Another relative issue is the teacher's interference degree in a learning activity. According to Mitra's work [14], children can achieve a satisfactory learning level by working themselves in small

978-1-4799-4038-7/14 $31.00 © 2014 IEEE
DOI 10.1109/ICALT.2014.225

teams helping each other. On the other hand, Collins et al. (1989) argue that in an apprenticeship perspective, students are able not only to appreciate the learning context but also actively use it under a variety of conditions [26]. Technology – based learning environments could support many apprenticeships principles like authenticity. This way, they could provide effective learning activities that support programming context. Several other approaches like simulation, problem based learning etc. could also be very helpful. Pedagogy of programming is a field that needs further extended research as its benefits for the young people looks to be great.

It is obvious that researchers should help students and also teachers and offer them the right tools and the knowledge as well as pedagogical directions in order to succeed in learning / teaching programming. Many summer camps [1] or after school activities [24] etc. for students from different ages and also for teachers [3] are being organized. It looks like there is a necessity to keep trying on this field.

III. PRIOR WORK AND RESEARCH QUESTION

Introducing programming to novice students is a very challenging attempt. The most of the research, that has been done till now, is about university introductory courses (freshmen). It is also useful to introduce and engage younger students in this field. Many researchers design workshops, summer camps or after school activities in order to achieve this goal and also study the results in several ways.

An interesting approach is the two week summer game camp [2] for students combined with a four week high school teacher professional development course using the Greenfoot Programming tool. Students' and teachers' survey results [2] showed improvement in computer knowledge and self confidence. It is very crucial not only to increase the students' interest but also to help teachers in their attempt to teach programming. Another workshop [9] program took place in Norway in order to introduce computer science and programming to twelve years old students. It offered an authentic experience in programming with scratch and using the Arduino hardware platform and recycling materials. The evaluative feedback from the students [9] showed that creativity has the potential to increase participation in computer science. In particular, it seems like creativity motivates students and increases their interest. Grover and Pea, while describing and evaluating a programming approach with mobile apps in App Inventor for Android [11], they underline the need of applying the learning theories in order to design the appropriate curricula and pedagogies in teaching computer science content

Up to now, researchers have mostly focused on developing visual programming environments and dealing with the problem of computer science acceptance from novice students. Both issues are crucial because of their possible influence in developing computational thinking and creativity. However, there is more to be done in the research area of learning and deeply understanding computer science concepts and particularly programming in a creative and technology – based way. Under this perspective, this project focuses on the following research question challenge:

RQ: *How can young students learn programming in an engaging way?*

In order to investigate many parameters of it, the following sub questions (SQ) have been formulated as a part of this research.

SQ1: *Does constructionism based teaching techniques have a significant positive impact on students' attitudes and effectiveness on learning programming?*

SQ2: *Which parameters of a programming environment have positive effect on young students learning?*

SQ3: *Does apprenticeship approaches can significantly contribute on learning programming activities?*

This research is expected to contribute to the development of meaningful ways and technologies to assist computer science and programming education. More detailed, our subject area is to explore the characteristics and the effectiveness of constructionist and apprenticeship methodologies and techniques while they are implementing in CS subjects. We will try to empirically evaluate them using a variety of visual programming tools in the typical school environment. Finally, we will try to empirically evaluate the teacher's degree influence in a learning activity.

IV. RESEARCH METHODOLOGY

Programming pedagogy including current teaching techniques and tools could have significant influence to young students. Our first step was to develop an understanding of the subject area conducting an exploratory research. In order to study many parameters deeper, we'll follow iterative cycles.

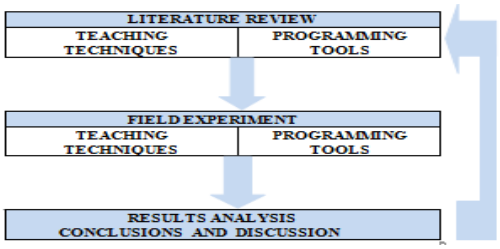

Fig. 1: Iterative Cycles of the Applied Methodology

These cycles will include literature survey, field experiments, results analysis, conclusions / discussion for various teaching approaches and programming tools. The experimental approaches will be applied in different ages and learning activities with various visual or tangible programming tools using teaching techniques based on constructionism or apprenticeship. Several scenarios will be developed in order to examine the research questions

according to the state of the art and the needs of the participated students, teachers and schools. Our findings will be analyzed with quantitative and qualitative methods. More detailed, we will use code analysis techniques before and after the learning activities. The results will be confirmed from our observations, pre and post questionnaires and interviews from the participants. Using the above techniques, we'd like to study our approach's effectiveness in compare with attitudes like enjoyment, creativity, usefulness etc. Finally, we will use our findings combined with more literature research as a new starting point in order to extend our research.

V. CONCLUSIONS

Engaging young students to programming might be a very useful and challenging issue. Various programming environments have been developed in order to succeed in the problem of computer science acceptance [17], [21]. Nevertheless, more research should be done and empirically evaluated on how instructional media like Scratch, Greenfoot or Arduino platform can be successfully implemented in the typical school environment of Primary and Secondary Education.

ACKNOWLEDGEMENTS

I'd like to kindly thank my supervisors Konstantinos Chorianopoulos (Ionian University, Department of Informatics, Corfu, Greece, choko@ionio.gr) and Michail N. Giannakos (Norwegian University of Science and Technology/Old Dominion University, mgiannakos@acm.org) for their support in my effort.

.

REFERENCES

[1] J.C. Adams, Scratching middle schoolers' creative itch, *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, ACM, 2010.

[2] M. Al-Bow, et al., Using game creation for teaching computer programming to high school students and teachers, *ACM SIGCSE Bulletin*, vol. 41, no.3, 2009, pp 104-108.

[3] A. R. Basawapatna, K. H. Koh, and A. Repenning, Using scalable game design to teach computer science from middle school to graduate school, *Proceedings of the Fifteenth Annual Conference on Innovation and Technology in Computer Science Education*, ACM, 2010.

[4] S. Beyer, K. Rynes, and S. Haller, Deterrents to women taking computer science courses, *Technology and Society Magazine*, IEEE, vol. 23, no.1, 2004, pp 21-28.

[5] J. Black, et al, Making computing interesting to school students: teachers' perspectives, *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education*, ACM, 2013.

[6] G. Demo, B. G. Marcianò, and S. Siega, Concrete programming: Using small robots in primary schools, *Advanced Learning Technologies*, 2008, ICALT'08, Eighth IEEE International Conference on, IEEE, 2008.

[7] D. Franklin, et al., Animal tlatoque: attracting middle school students to computing through culturally-relevant themes, *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education*, ACM, 2011.

[8] M. N. Giannakos, and L. Jaccheri, Designing creative activities for children: the importance of collaboration and the threat of losing control, *Proceedings of the 12th International Conference on Interaction Design and Children*, ACM, 2013.

[9] M. N. Giannakos, L. Jaccheri, and R. Proto, Teaching Computer Science to Young Children through Creativity: Lessons Learned from the Case of Norway, *Proceedings of the 3rd Computer Science Education Research Conference on Computer Science Education Research*, Open Universiteit, Heerlen, 2013.

[10] S. Grover, and R. Pea, Computational Thinking in K–12 A Review of the State of the Field, *Educational Researcher*, vol. 42, no.1, 2013, pp 38-43.

[11] S. Grover, and R. Pea, Using a discourse-intensive pedagogy and android's app inventor for introducing computational concepts to middle school students, *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, ACM, 2013.

[12] V. Isomöttönen, A. J. Lakanen, and V. Lappalainen, K-12 game programming course concept using textual programming, *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education*, ACM, 2011.

[13] Y. B. Kafai, Playing and making games for learning instructionist and constructionist perspectives for game studies, *Games and culture*, vol.1,no.1, 2006, pp 36-40.

[14] S. Mitra, and V. Rana, Children and the Internet: Experiments with minimally invasive education in India, *British Journal of Educational Technology*, vol.32, no.2, 2001, pp 221-232.

[15] M. Papastergiou, Digital Game-Based Learning in high school Computer Science education: Impact on educational effectiveness and student motivation, *Computers & Education,* vol.52, no.1, 2009, pp 1-12.

[16] S. Papert, and I. Harel, Situating constructionism, *Constructionism*, 1991, pp 1-11.

[17] M. Resnick, et al., Scratch: programming for all, *Communications of the ACM*, vol.52, no.11, 2009, pp 60-67.

[18] L. K. Soh, et al., Renaissance computing: an initiative for promoting student participation in computing, *ACM SIGCSE Bulletin*, Vol. 41, No. 1, ACM, 2009.

[19] A. Sperling, and D. Lickerman, Integrating AI and machine learning in software engineering course for high school students, *Proceedings of the 17th ACM Annual Conference on Innovation and Technology in Computer Science Education*, ACM, 2012.

[20] R. T. Tai, C. Q. Liu, A. V. Maltese, and X. T. Fan, Planning early for careers in science, *Science 312*, (5777), 2006, pp 1143-1144.

[21] I. Utting, et al. Alice, greenfoot, and scratch--a discussion, *ACM Transactions on Computing Education (TOCE)*,vol. 10, no.4, 2010, pp 17.

[22] A. Wagner, et al, Using app inventor in a K-12 summer camp, *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, ACM, 2013.

[23] D. C. Webb, A. Repenning, and K. H. Koh, Toward an emergent theory of broadening participation in computer science education, *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, ACM, 2012.

[24] U. Wolz, et al, Computational thinking via interactive journalism in middle school, *Proceedings of the 41st ACM technical symposium on Computer Science Education*, ACM, 2010.

[25] http://www.develop-online.net/news/uk-govt-outlines-computer-science-curriculum/0115853.

[26] J. Brown, A. Collins, and S. Newman. Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics, *Cognition and instruction: Issues and agendas,*1989, pp 453-494.