
AC 2011-1715: 3RD GRADERS EXPERIENCE ON USING AN AUTODI- DACTIC PROGRAMMING SOFTWARE: A PHENOMENOLOGICAL PER- SPECTIVE

Aikaterini Bagiati, Purdue University, West Lafayette

Having acquired a Diploma in Electrical Engineering and a Masters Degree in Advanced Digital Communication in the Aristotle University in Greece, and after having worked as an educator both in formal and informal settings for 10 years now, I have, since January 2008, started the PhD Program in the School of Engineering Education at Purdue University. My research interests are : Developmental Engineering; Engineering in PreK-3; Early Engineering Curriculum Development; Use of Art to enhance Engineering Design; Educational Software; Educational Robotics

Chrysoula Florou, University of Thessaly, Greece

Chrysoula Florou received the Diploma degree in computer & communication engineering from Department of Computer & Communication Engineering, University of Thessaly, Greece, in 2009. She's now an MSc student in the same Department.

Her research interests lie in the areas of technology enhanced learning, Web 2.0 applications for distance learning, and virtual learning environments.

Ioanna Kosmopoulou, University Of Thessaly, Greece

Prof. Elias N Houstis, Univ. of Thessaly

E.N. Houstis is currently a full Professor of Computer Engineering and Communications department at University of Thessaly, Greece, Director of Research Center of Thessaly (CE.RE.TE.TH.), and Emeritus Professor of Purdue University. USA. Most of his academic career is associated with Purdue University. He has been a Professor of Computer Science and Director of the Computational Science & Engineering Program of Purdue University. He is a member of working groups WG2.5 IFIP on mathematical software and European ICT Directors.

3rd graders experience on using an autodidactic programming software: A phenomenological perspective

Introduction

Interest in informal, meaning out of classroom, learning settings as an emerging research area is increasing¹ and although this is not a fully explored area, in terms of educational potential and settings², the benefits are expected to be large. One of the reasons for increased interest in informal learning is that with the advent of internet, a plethora of new and open educational resources has become available online. This has resulted in a completely different way for teachers, students and parents look for new content³. This is all happening at a time when the rapid technological advancement makes clear the need to identify optimum times and methods for technology and computer education, starting at early ages.

Besides today's need for technologically literate citizens, the fact that after 50 years programming is still considered to be a difficult course demands the location of a new, probably earlier, starting point, and different methods and media to teach it. "Programming is a very useful skill and can be a rewarding career. In recent years the demand for programmers and student interest in programming has grown rapidly, and introductory programming courses have become increasingly popular."⁴ Since the early 70's there has been an intense effort towards determining and adopting appropriate and efficient methods that would optimize the learners' ability to understand and solve programming related problems.^{4, 5} For these reasons, identifying the optimum way to teach programming and algorithmic thinking has for decades been one of the biggest challenges in the disciplines of software engineering and computer science. The attempts to address this challenge have led to introducing algorithmic thinking and presenting programming through microworld programs to early education classrooms.⁶

This paper presents an investigation to understand children's experiences when using a self-teaching interactive programming application. Developed by our research group, the application presents fundamental programming concepts and algorithmic thinking to early elementary school students. Since the intent of this application is to be used in a variety of learning settings, including informal home settings in which students will explore programming on their own, we sought a new student-oriented teaching application that provides the possibility for age appropriate self-assessment. The teaching approach is based on parallel use of the self-teaching application and use of the "Scratch" programming platform. This approach allows for interaction and experimentation with both fundamental and advanced programming concepts. The applications is interactive, provides constant feedback and, in parallel with teaching, offers the student developmentally appropriate rubrics for self-assessment.

The curriculum proposed through this application consists of ten educational modules that address fundamental programming concepts including algorithmic thinking, well-structured problem solving, assignment of variables, logic diagram creation, testing and debugging, and the use of sequential, conditional and repeated instructions. For developmental reasons the aforementioned terms are not labeled as such throughout the application. Children experientially address them when going through the proposed game scenarios. Children are instructed by the application to follow the ten modules in the order they are presented. Using developmentally

appropriate language, every educational module starts with a sketch of a child explaining, through an audio file, the concepts to be learned by the children. Using the Scratch platform, children then follow step-by-step instructions to create a small game. After completing this game they are given a similar scenario, addressing the same programming concepts, and are asked to make a similar game, this time on their own. After the completion of each module children are asked to respond to the self-assessment rubric. Throughout this process the application is keeping a score that is not visible to the student. Depending on the student's progress the application guides him/her to the next educational module or, if their score was found unsatisfactory, redirects them to a game that reinforces the same programming concept just taught. The student still has the flexibility to skip a module and proceed to a next one, or revisit a previous one.

Research Questions

The research questions examined in this study are:







1. How did an early education student experience fundamental programming learning and development of problem solving skills using the autodidactic software?
2. In what way and up to what point did this autodidactic programming software manage to work as an autonomous autodidactic learning tool?

Method

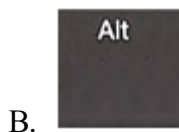
Data Collection

The autodidactic application was presented and used in a public elementary school located in a city in Greece. Based on convenience-sampling, participants in the study included twelve 3rd grade students, five boys and seven girls. Students used the application in a series of ten after class sessions that took place three times per week. An initial age appropriate questionnaire (see Table 1) was given to the students. In order to plan an appropriate introductory course, answers were used by our researchers to acquire a better understanding of the student's computer literacy level. The sessions started with the two researchers providing the students with an introductory course on how to use the self-teaching application. Following that, students used the application on their own. Data collection included the researchers' field notes, questionnaires provided to the students after completion of the third (see Table 2) and fifth (see Table 3) modules, and interviews with eight children after completion of the third, fifth, and final modules.

Table 1. Sample of initial questionnaire translated in English

I know how to turn on / shut down a computer
 
I know the basic parts of a computer (screen, mouse, keyboard, tower)
 
I know how to browse on the Internet
 

Which key do I have to use to make my keyboard write in capitals? Chose the correct answer



Match the text with the icon



Close window



Minimize window



Maximize window

There is a computer at home



There is Internet connection at home



Table 2. Sample of the questionnaire given to the students after the completion of the 3rd module translated in English.

How does Scratch seem to you so far?

A. Interesting

B. Not interesting

Did you manage to make the games you were asked to?



Do you know how to run Scratch?



Do you know how to run the readymade games?



Do you know how to insert more than one object on the Scratch scene?

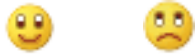


Table 3. Sample of the questionnaire given to the students after the completion of the 5th module translated in English.

Did you manage to make the games you were asked to?



Do you know how to run the readymade games?



Did you manage to make the game that makes your name flash?



Do you want to learn how to make more games with Scratch?



How do you find this Internet application? Put your answer in a circle

A. Interesting B. Not interesting

Would you like this Internet application to have more images?



Would you like this Internet application to have more children speaking?



Data analysis

Questionnaires were used to measure frequencies or responses. Knowing that the sample number of 12 students could not provide significant statistical power, results have only been used by the researchers as indicators towards their final finding's interpretations, and not as solid numbers that could support quantitative based arguments, and this is the lens through which they are further presented in this paper. Qualitative analysis using open coding under a phenomenological framework followed the data collected from the children's interviews.

Findings

During the interviews children were asked a series of open-ended questions regarding their experience using the self-teaching programming application. Four different perspectives emerged from the children's interviews regarding their experience while using the programming software (see Table 4).

Table 4. Categories that emerged from the children’s interviews regarding their experiences while using the self-teaching programming software.

Usability	Easy/difficult to use Efficient Clear to understand Easy/difficult to memorize
Interactivity	Navigation Feedback
Collaborations	Work on your own Work with your peers
Acceptance	Interesting/Boring Fun

In regards to how and the extent to which the interactive application managed to work as an autonomous autodidactic learning tool the following responses emerged from the questionnaires and the children’s interviews.

- Besides the use at school, four out of twelve students reported having accessed the Internet application from home on their own during their free time without facing any usability problems. They also reported success when trying to work on making games on their own.
- Students mentioned using modules of their choice rather than following a linear path that was prescribed for them.
- Students mentioned they had the chance to choose if they would move to a next module or revisit modules they had visited in the past, having at the same time the chance to also implement game scenarios they had thought of on their own, which allowed for personalization of their learning experience.
- The virtual characters (young boys and girls) used by the software to present the theoretical portions to the children were reported to have a significant effect on the children’s navigation, motivation, and retention of interest in the self-teaching application.
- In those instances where children had not succeeded in completing the game, the self-assessment rubrics were reported to have offered very detailed feedback. Children reported that this assisted them in identifying their own mistakes and in locating the exact step of the programming process where the mistakes had occurred.
- Based on the children’s self-assessment rubric data, ten out of twelve students were consistently able to complete the games task and proceed to further educational modules, while two out of twelve were redirected to repeat an educational module in a total of three different cases.

- One out of twelve students reported facing difficulties while filling out the self-assessment rubric due to an inability to remember the steps followed while working with Scratch.
- One out of the twelve students reported difficulties in understanding what the virtual boys and girls were saying. Two out of twelve children reported the gaming scenarios presented were slightly boring after some point.
- Eleven out of twelve students reported to have felt they had acquired the knowledge needed to be able to perform similar programming tasks and create similar games on their own

Besides children's responses, the following observations regarding children's experiences emerged from the researchers' field notes. Students appeared to find the self-teaching approach easier and gained familiarity with the software quickly. At the same time, the more familiarity they felt they were gaining as a group, the more they preferred to replace the researchers with their peers when it came to having a technical or content related question. When it came to accomplishing the tasks, three different learning strategies were identified: one group of children worked with the application only at school; a second group of children used the application both at school and at home; and a third group of children started using the application at school, but proceeded to work on parallel activities with their peers forming a parallel network gaming atmosphere. Discussing enjoyment, students mentioned they preferred this to other educational software they had seen in the past because of specific developmentally appropriate audiovisual features used by the application.

Conclusions

Taking a look at the findings, the developmentally appropriate audiovisual features used in the application appear to have been the strongest motivational factor for the children to get initially interested in following the curriculum, but the flexibility that it provided during use was probably the factor that helped maintain the children's further use of the software. Instead of using a linear path, children chose to move within modules according to their interest at the time; all but one child proceeded with developing their own scenarios at school or at home and, after getting familiar with the software, children went on working on their own machines while in parallel they were communicating as a group, all involved in playing the same game. The sum of these behaviors suggests that this open framework, in terms of content, work autonomy and availability for peer learning and peer challenging, is a successful learning framework that could be used to maintain young children's engagement and promote further interest in programming.

References

1. A. Bagiati, S. Yoon Yoon, D. Evangelou, I. Ngambeki.(2010) Engineering Curricula in Early Education: Describing the Landscape of Open Resources. *Early Childhood Research & Practice* V12(2)

2. Willoughby, T., Wood, E.(2008). Informal learning with technologies: opportunities and challenges – Introduction. In T. Willoughby& E.Wood (Eds.) *Children's learning in a digital world*. Malden, MA: Blackwell Pub.
3. Cape Town. “The Cape Town Open Education Declaration”. Retrieved October 2010 from <http://www.capetowndeclaration.org/read-the-declaration>
4. Robins, A., Rountree, J, Rountree, N. (2003) Learning and Teaching Programming: A Review and Discussion. *Computer Science Education*, V.13(2), pp137-172.
5. Soloway, E., & Spohrer, J.C. (Eds.). (1989). Studying the novice programmer. Hillsdale, NJ: Lawrence Erlbaum.
6. Shade, D., Watson, J. (1990) Computers in Early Education: Issues to Put to Rest, Theoretical Links to Sound Practice, and the Potential Contribution on Microworlds. *Journal of Educational Computing Research*. V6(4), pp375-392