Designing Creative Programing Experiences for 15 Years Old Students

Sofia Papavlasopoulou, Michail N. Giannakos and Letizia Jaccheri

Norwegian University of Science and Technology, Trondheim, Norway
michailg@idi.ntnu.no

Abstract. It is well known in the computer science education community that is important to encourage students to acquire programming skills and become creators and not only mere consumers. Different students have different needs and learning styles when introduced to programming and making activities, however it is challenging to accommodate all these needs while you design a workshop activity. In our approach we have designed and implemented a workshop program of 23 students' total, with the final goal of exploring and improving the design of appropriate workshops using the current learning environments. This paper presents an initial exploratory evaluation of a workshop program and the development of a set of guidelines for improving student experience. A set of best practices was developed through a focus group with experts using the technique of affinity diagrams. The results should be useful for designers and researchers who work with design and evaluation of programming workshop programs.

Keywords: Workshop program; design principles; creativity; programming; K-12 student

1 INTRODUCTION

Currently, several efforts to broaden participation in programming and introduce computational literacy to young students [1] [6] are in progress. Children interact with visual programming tools like Scratch [8] to learn how to code by creating interactive stories, games, animations, and simulations. Sesame workshop [9] has given new insights into how programming for children needs to be approached; in order to be both educational and entertaining. The process for achieving this mix relies on a development model that integrates expertise in media production, educational content (or curriculum), and research with children. Sesame Workshop philosophy [9] identify some of the challenges and solutions in designing interactive educational activities that can be used by children. Buechley et al. [1] argue that there is a need to make children programming a far more informal, approachable, and natural activity.

Although, programming activities for K-12 students have drawn great interest in the last years, little information is available on how to introduce computing literacy to pre-university students. Teachers and curriculum designers need to be aware and pay particular attention to any challenge students experience.

Make2Learn 2015 workshop at ICEC'15, September 29, 2015, Trondheim, Norway.

In this paper, we present our experience from a programming workshop focusing on K-12 students. With the knowledge extracted from this experience we aim to explore how any potential principles and recommendations can contribute to improve current practices and workshops. This paper focuses on our efforts to develop a programming workshop that will allow K-12 students to explore their potential interest in computer science education. Hence, we provide some first insights on: Principles for facilitating programming workshops for K-12 students.

To do so, we designed, implemented, and evaluated a programming workshop program. For the basic evaluation we employed response cards, where students write their feedback. After the workshop, we organized a focus group with two computing education researchers in order to organize the collected data.

2 MATERIALS AND METHODS

2.1 Workshop

The Norwegian University of Science and Technology offers six science programs for primary and secondary school students with the objective of introducing them and raising their interest to various science disciplines from physics, chemistry, mathematics, biology, energy, to computer science. The program, dedicated to computer science education is based on the hypothesis that the interactions between the young students and artifacts in a creative activity are vital. In this program (see figure 1) students introduced to programming by playfully interacting with digital artifacts that also exhibit physical and aesthetical characteristics. Such artifacts allow students to learn by iteratively testing and rebuilding their designs [3].

Fig. 1. Picture from one workshop: children play, program, interact with the assistants.

Programming concepts are introduced as needed for the progress of the development of the artifact. For example, we did not introduce and explain all possible loop constructs, but rather introduce each one when and if it is needed. Students experienced problems, but the problems did not frustrate students since the needed concepts have been introduced to them. The activity designed to be and finally flowed as an artifact development project. Each of the sessions was based upon a specific concept such as movement, sound and visual effect of the artifact. For each session, a set of tasks is presented to the students: make the artifact to move its hand, connect sensors input with character movements etc. By constructing programs to implement the tasks one after another, students ended up with constructing/participating in an "artifact development", while being taught different programming concepts. The workshop program was based on Scratch programming environment as well as different hardware like Arduino, sensors, motors etc.

2.2 Extracting Principles via Focus Group and Affinity Diagram Analysis

The main objective of our study is to perform an exploratory investigation of the programming workshop and justify the different principles, which are vital for students' experience. The first step of our methodology was to collect students' feedback. Hence, by the end of the workshop we asked students to fill a response card with specific activities and attributes helped them learn best, and additional over-all comments/recommendations related to the 2-day workshop program (see figure 2).

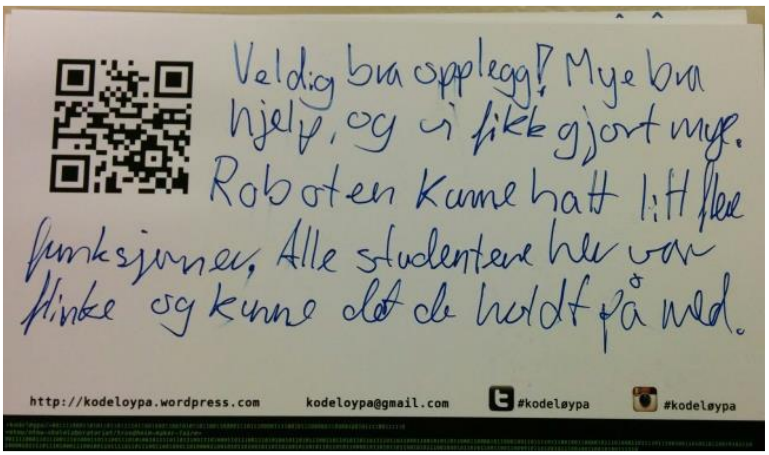

Fig. 2. Example of student's response card.

After cards collection a text analysis was performed and identified 73 comments/recommendations. Afterwards all the recommendations were translated into English, reprinted on to post-it notes and stuck onto the wall. Then, a focus group session was organized in order to analyze the gathered data. The purpose of the focus group, was to short all the comments/recommendations and form different principles. The focus group was consisting of two participants working in the area of computing education

research, not involved with the workshop; the objective was to organize all the collected data within an affinity diagram. The affinity diagram technique organizes the loads of data (students' recommendations in our case) to greater detail and often leads to results based on a consensus among participants [7]. This technique is appropriate to organize large amounts of qualitative data in groups according to the relationships among the ideas or topics. Affinity diagram technique includes tree main steps:

(1) Create notes for each idea
(2) Identify related ideas
(3) Categorize all notes in groups

As aforementioned all the post-it notes stuck onto the wall. Then, both participants review, group and reposition them within the different categories and tried to construct sub-categories, if possible. This was an iterative process that consisted of adding or removing post-its until a pattern was discovered. Finally, the participants made headings for the constructed categories and subcategories (see figure 3).

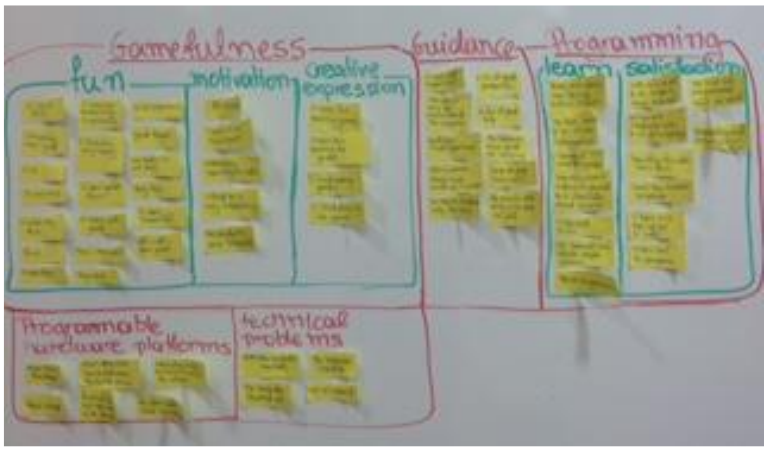

Fig. 3. The finalized affinity diagram: Overall recommendations categorization.

3 RESULTS

The affinity diagram consisted of the 73 recommendations and sorts them in five categories (figure 3): a) gamefulness, b) guidance, c) programming experience, d) programmable hardware platforms and e) technical problems. Initially, each category was consisted of 5-31 items. Then, the focus group indicated that within each of two general categories (gamefulness and programming experience) could correspond three subcategories. Also, six best practices were removed because were considered as irrelevant. The five general categories and their subcategories are described below.

Gamefulness: The "Gamefulness" category was defined as the use of game elements during the workshop with an aim to increase engagement and motivation. This

category includes three sub-categories (fun, motivation and creative expression) that clarify the different aspects of children's attitude to coding. "Fun" includes all practices/ ideas that describe the workshop as joyful experience. The "Motivation" subcategory was defined by the focus group as all the recommendations showing that coding was an interesting experience for students and most of them would like to participate in another workshop. Third sub-category, "creative expression" consists of practices/ideas that allow students' to express their creativity on coding, like making games controlling the artifacts.

Overall, almost all students agreed with the idea that they had fun during the workshop program. Programming workshops should aim to provide a pleasant atmosphere, giving the impression that programming is an enjoyable experience. Student's intention to participate again in other creative programming workshop increases when they feel happy during the workshop [4]. The attractive appearance of the digital artifacts with physical and aesthetical characteristics can be important for student's interest in programming. For example, artifacts should look like a character that students are familiar with and could support relevant play activities that student's can explore.

Guidance: In this category, all recommendations related to the importance of help and assistance during the workshop were sorted. As mentioned, each loop construct was explained only when and if it was needed. Some of the students seemed to be more familiar with the programming environment and other had more limited understanding. Students expressed their appreciation for the guidance and help on how to apply the different programming concepts in order to interact with the artifacts.

Some of the most important aspects of this category were collaboration and communication among the students. Also, peer support and guidance allowed students to become confident with programming. In summary, proper and sufficient guidance was very beneficial to help students to construct the appropriate competences during the workshop [2]. As aforementioned, programming workshops should provide a happy environment and students should feel free to ask and collaborate for a better "artifact development".

Programming Experience: In order to describe the general category of "programming experience" the focus group created two sub-categories. The first sub-category is "learn", which contains the recommendations related to the learning procedure of the workshop. For example, some of the comments mentioned "I learn about coding" indicating that the workshop achieved its goal. Also, recommendations that showed satisfaction from coding were sorted in the second sub-category "satisfaction". Many students liked very much constructing programs and execute the tasks successfully. Their satisfaction derived from the fact that they were able to complete the asked tasks, and construct the needed artifact.

It is also important to stress the benefits of students' satisfaction. The instructors noticed that satisfaction leaded students to minimize their frustration and follow the needed tasks. Therefore, it is crucial for the design of the workshop to adapt to different students' needs like age and previous knowledge.

Programmable hardware platforms: This category includes the recommendations related to the interaction with the physical components. Students have the freedom to explore how artifacts could move, communicate with the environment, make sounds, etc. The workshop is based on a visual programming environment to construct these affordances. The programming environment (Scratch for Arduino) requires that a physical artifact is the beneficiary of the developed software. Students felt more positively interacting with the artifacts, understand the functionalities and explore how they work. They had the opportunity to see in practice how different programming concepts are applied.

Students participated in an "artifact development" using computing tools and techniques for creative expression. Interaction with the artifacts requires flexible hardware and software tools. These tools could be implemented to specific disciplines from physics, chemistry and mathematics to poetry, history and human anatomy [7]. Educators should connect computational artifacts development with other disciplines. The variety of different disciplines ensures that students' interest will be raised, by connecting programming with other well-known to the students' notions like scientific phenomena in physics and chemistry.

Technical problems: In this category the focus group sorted all recommendations that describe problems with the software or the artefact. For example when the computer crashed, or took a lot of time to perform a task; another example is wrong connections with the boards, functional sensors etc.

This category stresses the importance of a robust software and hardware environment to ensure an uninterrupted progress as well as to support students' creativity and imagination.

4 CONCLUSIONS AND THE WAY AHEAD

In this paper we presented the results from an investigation of a programming workshop for K-12 students. Our results provide an initial attempt to exploit knowledge from K-12 students and model this knowledge into useful principles for educators and curriculum designers who aim to develop K-12 programming workshops. The study described in this paper has led to a set of guidelines for improving and better designing programming workshops. The guidelines were backed by students' experience and have been exposed to several stages of validation and organization (focus group, affinity diagram analysis), which should provide some assurance of their validity. Based on this, five have been extracted.

The main principle for facilitating programming workshops for K-12 students is to provide a pleasant atmosphere. Students' interest rises when they have the feeling that they can playfully interact and explore the functions of the digital artifacts. Educators and curriculum designers should offer practical applications in order to empower students' interest to programming. They should aim to improve the overall learning procedure of the workshops by reforming the digital artifacts, giving the proper guidance

and define the specific goals. This will offer perseverance and reinforce students' passion to deal with challenges, failures, adversity and success with computer science.

We want to emphasize that our findings are preliminary with inevitable limitations. Our future research will concentrate on further refinement of the proposed principles by applying and evaluating them on real conditions. Furthermore, educators, practitioners and researchers in the areas of computer science education should evaluate the proposed principles in order to ensure their understanding and seek suggestions and extensions.

Acknowledgements. We would like to thank all the participants of the study. We also thank Finn Inderhaug Holme, Irene Dominguez Marquez and Ilse Gerda Visser for setting up and running the workshop, P. Bøyesen and A. Eriksen I. for their contribution in the early phases of this project, Eirik Høydalsvik for assisting us with the translation of the cards and the colleagues at the department who work for making Kodeløypa possible.

5 References

1. Buechley, L., Eisenberg, M., Catchen, J., and Crockett, A. 2008. The LilyPadArduino: Using computational textiles to investigate engagement, aesthetics, and diversity in computer science education. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy April 5–10, 2008) CHI'08. ACM, New York, NY, 423–432 DOI= http://doi.acm.org/ 10.1145/1357054.1357123.
2. Buffum, P. S., Martinez-Arocho, A. G., Frankosky, M. H., Rodriguez, F. J., Wiebe, E. N., and Boyer, K. E. 2014. "CS principles goes to middle school: learning how to teach "Big Data". In *Proceedings of the 45th ACM technical symposium on Computer science education* (Atlanta, Georgia, USA March 5–8, 2014) SIGCSE'14. ACM, New York, NY, 151-156 DOI= http://dx.doi.org/10.1145/2538862.2538949 2014.
3. Cassell, J. 2004. Towards a Model of Technology and Literacy Development: Story Listening Systems, *Journal of Applied Developmental Psychology* 25, 75-105.
4. Giannakos, M. N., Letizia, J., and Leftheriotis, I. 2014. Happy Girls Engaging with Technology: Assessing Emotions and Engagement Related to Programming Activities." *Learning and Collaboration Technologies. Designing and Developing Novel Learning Experiences*. Springer International Publishing, 398-409.
5. Hamner, E., and Cross, J. 2013. Arts & Bots: Techniques for distributing a STEAM robotics program through K-12 classrooms. In *Proceedings of the Third IEEE Integrated STEM Education Conference, Princeton, NJ, USA*.
6. Kafai, Y. B., Lee, E., Searle, K., Fields, D., Kaplan, E., & Lui, D. (2014). A crafts-oriented approach to computing in high school: Introducing computational concepts, practices, and perspectives with electronic textiles. *ACM Transactions on Computing Education* (TOCE), 14(1), 1.
7. Maguire, M., and Bevan, N. 2002. User requirements analysis. A review of supporting methods. In *Proceedings of IFIP 17th World Computer Congress*, (Montreal, Canada), 25-30.
8. Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K.. and Kafai, Y. 2009. Scratch: programming for all. *Communications of the ACM*, 52(11), 60-67.

9. Revelle, G.L. 2003. Educating via entertainment media: The Sesame Workshop approach. *Computers in Entertainment (CIE) - Theoretical and Practical Computer Applications in Entertainment* ACM Comput. Entertain. 1(1), Art. 7.