Impacts of Block-based Programming on Young Learners’ Programming Skills and Attitudes in the Context of Smart Environments

Mazyar Seraj
Institute of Computer Science, University of Bremen, 28359 Bremen, Germany
Cyber-Physical Systems, DFKI GmbH, 28359 Bremen, Germany
seraj@uni-bremen.de, mazyar.seraj@dfki.de

ABSTRACT
In computer programming education, showing the application of programming in reality has become a common way to introduce it to young learners. However, we have limited knowledge of how best to utilize smart objects and environments to foster the learners’ programming skills and develop a positive attitude towards programming. My research focuses on filling this gap by presenting an educational block-based programming tool that brings together the hot topic of smart environments and the visual programming paradigm. The end goal is empirically investigating the impacts of state-of-the-art smart technologies together with block-based programming on the learners’ programming skills and attitudes towards programming in non-formal learning environments.

CCS CONCEPTS
• Applied computing → Computer-assisted instruction; • Social and professional topics → Computer science education;

KEYWORDS
block-based programming, young learners, smart homes, experiment, programming skills, attitudes towards programming

1 CONTEXT AND MOTIVATION
Inexperienced and young learners typically have difficulties with respect to the programming experiences and activities. These difficulties are mainly because of their lack of syntactic knowledge, conceptual knowledge and strategic knowledge [12]. Considering the complexity of introductory programming for the learners, visual programming has become more and more popular [18]. In particular, block-based programming have emerged as an area of active research.

Block-based programming environments have become the standard medium of instruction in the design of introductory programming courses for young learners [18]. As a result, they are employed by researchers and educators to enable the learners to learn programming and author computer programs. In addition to these programming environments, an interesting and motivating context is needed to encourage young learners to start with programming activities. Scientific works emphasize that tangible interactive objects benefit learning, specially for young learners [3, 8]. Moreover, countless block-based programming environments have been employed together with these objects in order to improve the learners’ emotional engagement, attitudes, and their computer programming performance [3, 8]. Nevertheless, we have lack of investigation on the impacts of new and powerful technologies (which provide possibilities to tightly connect computer science to reality and introduce the future) on young learners’ programming skills and attitudes. My research is aimed at better understanding of how the use of block-based programming together with state-of-the-art smart technologies can leverage young learners’ interest in programming, and support the acquisition of programming skills.

2 BACKGROUND AND RELATED WORK
One important feature in learning programming, especially for young learners, is to enable them to understand how relevant programming and computer science are to their daily life [2, 8]. Being rooted in constructionist learning theory programming tangible objects has a long history in education [1]. Meanwhile, countless programmable kits and computational textiles are on the market and have entered into educational institutions [2, 8]. With respect to constructionist learning theory, young learners can learn better when they design and construct interactive and tangible objects that are personally meaningful to them, such as computational textiles, robots, and interactive objects [8]. Thus, researchers and educators designed introductory programming environments to support acquisition of programming skills through designing and creating visible and tangible objects. In both the Computer Science Education (CSE) and Computer-Human Interaction (CHI) research communities, several scientific studies tried to investigate various forms of smart devices (e.g., tangible artifacts [3, 5] and robots [9, 10]) in order to motivate young learners and show them how modern technologies relate to their daily life. However, relatively little attention has been devoted to the potential of smart homes to support the claim, which they have a direct impact on young learners’ performance and attitude towards programming.

Nowadays, physical computing technologies and programming environments with low barriers have been established [11]. These technologies and environments have been used to explore computational concepts by building smart tangible objects in educational contexts. However, the application area of designing smart objects can be extended to the state-of-the-art area of smart environments. From a technical point of view, smart environments comprise networking, intelligent control and home automation of key electrical appliances and services [4]. From a user perspective, a smart environment is a space equipped with smart technologies to enhance the quality of its inhabitants’ life by providing services that control, monitor and support their well-being [7]. Social relevance of computing has been identified as attracting, especially for women, to participate in computer science education and society [6]. Hence, my research focuses on the theme of smart environments which...
offers high potential for a purposeful application in computing education that is meaningful for diverse target groups of learners.

3 PROBLEM STATEMENT

There is much to show on teaching programming via block-based programming and tangible interactive objects to young learners. In an effort to address this intervention, scientific research has been done on the effectiveness of using these platforms in order to teach basic programming concepts. However, less is known about how inexperienced and young learners’ performance and attitude towards programming are influenced over time in the context of real life-size smart environments, such as smart homes. Unfortunately, these environments are not accessible for young learners due to relying on modern and powerful technologies. Therefore, relatively little attention has been given to show potential for using educational block-based programming environments to make state-of-the-art smart technologies accessible for the learners. Thus, my dissertation research focuses on answering the following two-part research question:

How do young learners’ programming performance and attitude change over time in non-formal programming training sessions with respect to using block-based programming and smart homes as a medium of teaching programming?

4 RESEARCH GOALS AND METHODS

My research explores educational block-based programming environments in the context of smart objects and environments to achieve two main objectives. First, exposing young learners to programming activities to help them to realize that computer programming can be presented in a way which is not necessarily difficult to understand. Second, utilizing the results of these programming activities to effectively develop the learners’ basic programming skills and engage them in future learning computer programming.

The exposure of young learners to programming activities focuses on presenting an educational block-based programming tool that brings together the hot topic of smart environments and the visual programming paradigm [13, 17]. In order to illustrate the effectiveness of our approach, it is employed to design one-day non-formal programming training sessions in the context of smart homes [14, 15]. Furthermore, to offer insights into the impacts of embedding the construction of smart objects in context of smart environments, longer period (2- to 4-day) of training sessions are conducted [16]. Each training session is divided into two parts: (i) introduction to basic programming concepts, and (ii) implementation of these concepts on tangible objects and construction of a smart object in the context of smart homes. This helps us to explore the learners’ programming performance and their attitudes towards programming over time. In this respect, programming tasks can be more diverse and complicated for both groups of learners (with and without prior programming experience), and both genders (boys and girls). In this phase, we focus more to find out the learners’ trajectories of (i) attitudes towards programming (in terms of confidence, interest, and enjoyment), (ii) acquisition of programming skills, and (iii) programming experience (in terms of ease-of-use, ease-of-learning, usefulness and satisfaction), using the block-based programming environment and smart tangible objects.

5 CONTRIBUTIONS

In comparison to the previous work, the main contribution of the proposed approach are as follows:

- Using the programming environments in order to help the learners to solve programming problems in the context of real-world environments and tangible objects.
- Development, and evaluation of non-formal training sessions to teach basic programming concepts to the learners, and to arouse their interest in programming.
- Showing how the learners’ programming performance and attitude change over time, when they apply their new gained programming skills in a tangible object and make it smart.
- Studying the construction process of smart tangible objects as interactive artifacts with physical computing material.

6 ACKNOWLEDGMENTS

I would like to thank my advisor, Prof. Rolf Drechsler, for his continuous support and guidance. I deeply appreciate the opportunity he gave me to be part of his research groups in DFKI and AGRA.

REFERENCES

[1] Paulo Blikstein et al. 2015. Computationally enhanced toolkits for children: historical review and a framework for future design. Foundations and Trends® in Human-Computer Interaction 8, 1–2 (2015).
[2] Leah Buechley, Mike Eisenberg, Jaume Cachet, and Ali Crockett. 2008. The LilyPad Arduino: using computational textiles to investigate engagement, aesthetics, and diversity in computer science education. In Proc. of the SIGCHI conf. on Human factors in computing systems. 423–432.
[3] Brittany Garcia, Sharon Lynn Chu, Beth Nam, and Colin Banigan. 2018. Wearables for learning: examining the smartwatch as a tool for situated science reflection. In Proc. of the 2018 CHI Conf. on human factors in computing systems. 1–13.
[4] Li Jiang, Da-You Liu, and Bo Yang. 2004. Smart home research. In Proc. of 2004 international conf. on machine learning and cybernetics, Vol. 2. IEEE, 659–663.
[5] Eva-Sophie Katterfeldt, Nadine Dittert, and Heidi Schellhammer. 2009. EduWear: smart textiles as ways of relating computing technology to everyday life. In Proc. of the 8th International Conf. on Interaction Design and Children. 9–17.
[6] Nazih Zaman Khan and Andrew Luxton-Reilly. 2016. Is computing for social good the solution to closing the gender gap in computer science?. In Proc. of the Australasian Computer Science Week Multiconference. 1–5.
[7] Davit Markryan, Sovvas Papagiannidis, and Efthelarios Alamanos. 2019. A systematic review of the smart home literature: A user perspective. Technological Forecasting and Social Change 138 (2019), 139–154.
[8] Alexandros Merkouri, Konstantinos Chorianopoulos, and Achilles Kameas. 2017. Teaching programming in secondary education through embodied computing platforms: Robotics and wearables. ACM Transactions on Computing Education (TOCE) 17, 2 (2017), 1–22.
[9] Joseph E Michaelis and Bülle Mutlu. 2019. Supporting Interest in Science Learning with a Social Robot. In Proc. of the 18th ACM International Conf. on Interaction Design and Children. 71–82.
[10] Omar Mubin, Catherine J Stevens, Suleman Shaid, Abdulkhiril Al Mahmoud, and Jian-Jie Dong. 2013. A review of the applicability of robotics in education. Journal of Technology in Education and Learning 1, 209-9015 (2013), 13.
[11] Mareen Prezybilla and Ralf Romenke. 2014. Physical Computing and Its Scope: Towards a Constructionist Computer Science Curriculum with Physical Computing. Informatics in Education 13, 2 (2014), 241–254.
[12] Yizhou Qian and James Lehman. 2017. Students’ misconceptions and other difficulties in introductory programming: A literature review. ACM Transactions on Computing Education (TOCE) 18, 1 (2017), 1–24.
[13] Mazyar Seraj, Serge Autexier, and Jan Janssen. 2018. BESM, a block-based educational programming tool for end users. In Proc. of the 10th Nordic Conf. on Human-Computer Interaction. 886–891.
[14] Mazyar Seraj, Cornelia S Große, Serge Autexier, and Rolf Drechsler. 2019. Look what I can do: acquisition of programming skills in the context of living labs. In 2019 IEEE/ACM 41st International Conf. on Software Engineering: Software Engineering Education and Training (ICSE-SEET). IEEE, 197–207.
[15] Mazyar Seraj, Cornelia S Große, Serge Autexier, and Rolf Drechsler. 2019. Smart Homes Programming: Development and Evaluation of an Educational Programming Environment for Young Learners. In Proc. of the 18th ACM International Conf. on Interaction Design and Children. 146–152.
[16] Mazyar Seraj, Eva-Sophie Katterfeldt, Serge Autexier, and Rolf Drechsler. 2020. Impacts of Creating Smart Everyday Objects on Young Female Students’ Programming Skills and Attitudes. In Proc. of the 51st ACM Technical Symposium on Computer Science Education. 1234–1240.
[17] Mazyar Seraj, Eva-Sophie Katterfeldt, Kerstin Buh, Serge Autexier, and Rolf Drechsler. 2019. Scratch and Google Blockly: How Girls’ Programming Skills and Attitudes are Influenced. In Proc. of the 19th Koli Calling International Conf. on Computing Education Research. 1–10.
[18] David Weintrop. 2019. Block-based programming in computer science education. Commun. ACM 62, 8 (2019), 22–25.