The Effect of Tangible Augmented Reality Interfaces on Teaching Computational Thinking: A Preliminary Study

Anna Gardeli(✉) and Spyros Vosinakis

Department of Product and Systems Design Eng, University of the Aegean, Mytilene, Greece
{agardeli,spyrosv}@aegean.gr

Abstract. Teaching introductory computer science is gradually shifting from learning a computer programming language towards acquiring more generic computational thinking skills. At the same time, more emphasis is placed nowadays in natural and playful approaches that can potentially increase student motivation and engagement. One promising such approach is the combination of tangible elements with augmented reality technology, where the instructions can be given in the real world by manipulating physical elements, and the output is presented in a digitally enhanced space. Despite its potential, this approach has not yet been evaluated in formal educational settings. In this paper we present the results from a preliminary study in an elementary school that compared a tangible AR game with the same game in an unplugged version, to examine the effect of the interface on student motivation, effectiveness and teaching practice. The results indicate that a tangible AR approach can improve the engagement and collaboration of students in classroom activities and affects the role of the instructor compared to unplugged activities. The paper concludes with a number of open issues that need to be further studied.

Keywords: Computational thinking · Physical programming · Augmented reality · Unplugged activities

1 Introduction

Contemporary education gradually leans towards more intrinsically motivated learning methods, wherein students themselves are the main driving force behind their own academic explorations and learning experiences. Thus, enjoyment and motivation are becoming increasingly more important learning factors in both formal and non-formal educational settings [1]. Within this scope, researchers and educators are in search of ways to increase engagement, attract students' interest and challenge them to take part in learning activities through active collaboration, social interaction, physical actions and game-like activities, based on new, more appropriate technology [2].

In regard to computer science, efforts have been made in the last decade to apply new tools and methods in the classroom and to assess their effect on student motivation and engagement. At the same time, the educational focus has shifted from learning how to program in a given language towards acquiring more generic problem solving and

M. E. Auer and T. Tsiatsos (Eds.): ICL 2018, AISC 916, pp. 673–684, 2020.
https://doi.org/10.1007/978-3-030-11932-4_63

algorithmic thinking skills, an approach termed as ‘Computational Thinking’ [3]. The first and still most popular category of efforts towards this end is visual programming, an approach which is based on the manipulation of visual elements to create a program instead of typing written code, and on the visualization of the code execution [4]. This approach helps students avoid syntactical errors and usually results to the development of more engaging programs compared to traditional programming environments. A more recent category that is gaining popularity are the unplugged activities, i.e. problems and algorithms that are solved in the real-world using everyday objects, such as pen and paper, toys, fruit, etc. [5]. In this case, students become familiar with algorithms and computational processes before even using any digital device, thus making it easier to disassociate computational thinking from computers. Finally, another interesting trend is physical programming, where students create programs through the manipulation of physical objects. They combine tangible interactive objects to construct their algorithms and they can observe the results of their instructions in the physical (e.g. through robots) or digital space [6]. Physical programming platforms have shown strong indicators of promoting engagement and motivation by combining the positive aspects of unplugged activities with the advantages of digital technology.

In the recent years, augmented reality (AR) technology has provided novel affordances for educational activities by presenting a digitally enhanced version of the physical world, thus creating further opportunities for physical computing. A number of researchers have recently combined camera-based AR systems with tangible artifacts to create physical programming environments with visual feedback. The aim was to introduce an engaging and motivating collaborative approach for children and help them learn abstract concepts of computational thinking [7, 8]. AR technology is nowadays easily accessible and affordable through mobile devices, and as a medium for physical programming platforms it shows great potential for integrating in formal settings, as (a) it can be more inexpensive, (b) it can be used collaboratively by groups of students, (c) it can be portable and (d) it makes learning more engaging.

Despite the fact that the combination of physical computing with mobile AR seems to be a promising direction for novel educational environments for computational thinking, the number of studies regarding their effectiveness and their impact on student motivation is limited. Furthermore, and to the best of the authors’ knowledge, such systems have not yet been evaluated in formal educational settings.

The aim of our research is to study the effects of a tangible AR interface for teaching computational thinking in formal educational settings, regarding student motivation, learning effectiveness, collaboration and instructor support. We have designed and implemented a prototype tangible AR game for learning sequential instructions using mobile devices and 3D printed manipulatives, and performed a user study of its effectiveness during an educational intervention in a 3rd grade elementary school (8 to 9 years old). In the study we used two treatments: the experimental group played the tangible AR game, whilst the control group played the same game in an ‘unplugged’ version. The results of the study revealed some interesting findings regarding the suitability of the approach, its impact on student collaboration and on the role of the instructor, as well as some issues that need to be further studied.

2 Related Work

In recent years, a strong interest towards computational thinking (CT) is becoming apparent, regarding both education in the fields of STEM and computer science (CS), and students' overall development within modern society. Computational thinking refers to a specific way of understanding, analyzing and solving problems using known patterns, that can be directly and effectively visualized within the CS discipline but also apply to many different areas of science and everyday life. Instead of focusing on the syntax and semantics of computer programming, CT refers to the development of higher-level skills. However, it is a notion that has taken different dimensions over the years and the skills encompassing CT vary in the literature, as there is lack of a widely accepted definition. Critical thinking and problem solving have been reported as the most commonly related skills to CT [9]. Nevertheless, more skills have been correlated with CT in various studies, such as constructing algorithms, debugging, simulation and socializing [10].

Although many studies provide evidence that CT is more than a foundational skill in programming, implementing CT in typical education still faces many challenges. Some of those challenges have their roots in the core limitations of the current formal education settings, such as (a) the existing curriculum standards, (b) educators' lack of suitable skills and training and (c) limited institution resources which lack suitable equipment and infrastructures. As a result, much of the work in CT referring to K-12 education remains in out-of-school environments [11].

The most widely used approaches for teaching CT concepts are structured around students' interaction with computers, mainly based on block-based programming languages. A common approach in this direction is the use of visual tools, such as Scratch, where the program is written and its output is visualized in a graphical environment.

However, there is another approach gaining popularity in CT education, which includes activities with no use of digital devices: unplugged activities. Such activities are based on logic games using cards, strings or physical movements that represent computer science concepts such as algorithms or data transmission [12]. One of the main benefits of these activities is that they lowering the access barriers to the required equipment and infrastructures of educational tools [8], which results in easier implementation in field settings, consequently more concrete evidence concerning their use and effectiveness.

As technology evolves in exponentially increasing pace, block-based programming language in form of tangible interfaces have been gradually introduced in teaching CT, which enable programming by simply assembling physical blocks together. In this approach, each physical block stands for a respective command, and their combination creates a program. Students are programming on their desk or on the floor, which in the context of a classroom provides educators with the opportunity to more effectively determine and control the entire activity. Additionally, physical interfaces are exploiting the knowledge and skills that students already have through their own, non-computer related, experiences of everyday life, instead of forcing them to learn new skills, such as using commands of programming languages or non-physical interactions [13].

Physical programming platforms are following two approaches; the first include only physical parts and the second use the combination of digital and physical elements. One of the first examples of physical programming is the Programmable Bricks, created in the nineties at MIT [6, 14]. More recently, Google began its own research project within the field, with Project Blocks [14], which is based on Google visual programming language, Blockly.[1] This approach uses microcontrollers and microprocessors inside the blocks to communicate with the robot which increase the cost of construction. In the second approach, platforms like Tern [13], propose a solution where each physical block is read by an overhead camera, translated into commands, compiled by a computer and sent to a physical robot nearby. A more modern example is Osmo,[2] which uses the camera of an iPad device in order to execute the programming solution on screen with a digital character. This approach bears limitations concerning distance, light and color, but it uses widely available technology and reduce cost by combining the physical platform with a digital screen, gradually leading us to AR-based learning.

AR-based learning provides a new way to meet those challenges, concerning cost and complexity of the educational systems, while it could offer a natural way to present information based on reality [15], as it can enhance learning in terms of context visualization and learning interactivity [16]. AR allows augmented information for the surrounding real world to become interactive and manipulable [17], which could encourage students to have positive attitudes toward exploration, inquiry, logical thinking, and reasoning. In short, the two main affordances of tangible AR, i.e. visualization and interactivity [18], might have the potential to overcome some core difficulties that novices encounter in learning programming, such as abstraction and complexity. Furthermore, as a medium it may support the integration of modern technology in formal educational settings using widely available equipment.

A lack of studies is observed concerning concrete evidence on if, how and why tangible interfaces (TUIs) and more natural gestures create more effective learning experiences compared to other modern widely accepted approaches in formal education of computer science. Since now, there are cost and complexity issues, as the construction of tangible systems for programming is considered to be expensive, which provides an explanation for the lack of field evaluations [19]. Therefore, research in the field tends to work on how to make physical programming platforms more cost-effective, while enhancing interaction with widely available technology in formal education context [7, 8]. Mobile Augmented Reality (AR) technologies show great potential towards this direction.

[1] https://blockly-games.appspot.com/.

[2] https://www.playosmo.com.

3 A Tangible Augmented Reality Approach

We have designed and implemented "ALGO and his 'rithm", a collaborative game-based programming activity based on a widely accepted teaching model for CT concepts. The game-based learning approach is a shift from a traditional teacher-centered learning environment to a student-centered environment where students' participation and engagement are enhanced [20]. Our approach is based on mobile AR technology and is specifically focused in formal settings of primary education. This context of use significantly defined our design, as it creates a number of limitations and requirements. The core design decisions of our approach, in order to support learning and create an effective and controlled environment for both student and educators, are (a) the combination of tangible interaction with digital elements, (b) the collaborative context and (c) the low-cost and widely available equipment.

The educational goals of ALGO are to teach basic computational concepts, such as the development of algorithmic logic and sequential programming skills in a collaborative context. We address CT on the aspects of building algorithms and socializing within Computer Science discipline, as an introduction to programming for children. The main challenge introduced in the game is to create a programming solution – a sequence of steps – to assist a robot, named ALGO, collect a number of items and escape from a series of platforms (levels). "ALGO and his 'rithm" is following similar patterns regarding gameplay, challenges and educational goals to other games such as Light-Bot (2008)[3].

The game consists of two parts; the tangible part and the digital part. As every programming platform, there is the "programming language" (i.e. the commands) that is represented in tangible form and the execution that occurs on screen. It uses tangible blocks of commands in order to control ALGO on - screen, by taking advantage of marker – based AR using image targets. The user creates the programming solution of the given problem using the tangible blocks and then use the device to scan their solution and watch ALGO execute it on-screen (Fig. 1).

The prototype built for this study is limited to problems requiring sequential executions, which was the focus of our first study. The prototype is based on physical blocks of five different types, each of which represents a command that triggers ALGO's actions (to move in all four directions and to take an object). Respectively, the challenges introduced in the digital game are of the same complexity. In future implementation, we are planning to extend both the command set and the challenges to allow for conditionals and loops, and also to extend the gaming elements in the software.

The prototype has been implemented in the Unity game engine and uses Vuforia SDK which integrates AR capabilities in the game, in order to create a real-time interaction. The application runs on Android devices, tablet and smartphone. In our approach we have taken advantage of 3D printing as construction method for the tangible elements of the system, as they have been specially designed to snap together. However, the system is flexible regarding the physical elements, so it can support the

[3] http://lightbot.com/.

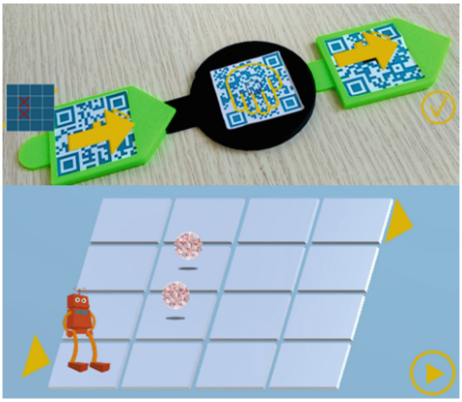

Fig. 1. The tangible AR version of "ALGO" game.

use of DIY tangible blocks using other materials like paper or cardboard and only the markers are required in order to play the game. That way our system supports an important aspect of this approach, as mentioned above, the requirement of widely available equipment and low-cost materials.

Before using our prototype in the main study, we have conducted a small lab session to evaluate its usability and improve any critical issues that might occur. For our evaluation we used six (6) students, age 8 to 9, forming two groups, where they had to complete three goals of the game. Our preliminary results showed that the interaction with the game was intriguing and challenging, as the participants did not have prior experience with AR, and indicated high levels of motivation and engagement for the kids. However, some technical and usability issues surfaced related to marker detection. The main problem was that the preliminary prototype supported linear "scanning" of the blocks (i.e. all blocks should be arranged horizontally), which made it difficult for the AR camera to detect them all simultaneously. This issue was resolved in the latest version of ALGO, by allowing for more than one lines of commands. Further minor usability issues that were observed during the tests, such as the size of specific virtual elements and the lack of audio or visual feedback for some actions, have also been improved.

4 User Study

This study is driven by the following research motivations: (a) what is the perceived value and effectiveness of using a tangible interface in an AR-based programming activity compared to an unplugged activity, in terms of programming performance and motivation, then (b) how the role of the educator is differentiated in class settings concerning those two approaches, in terms of involvement and engagement and finally, (c) does the AR-based approach support the educational goals, while addressing the limitations of the current formal education settings, such as educators' skills and training, and institution resources? What are the factors and the characteristics of the system that contribute to that?

We have performed a controlled experiment for addressing the above questions using the tangible AR approach as the experimental condition and an 'unplugged' version of the same game as the control condition. We used the same challenges (three tasks to be solved) in both treatments. In the experimental condition users interacted with the tangible AR prototype described in Sect. 3, whilst in the control condition they played the same game in an 'unplugged' version, by interacting only with physical elements. Specifically, the unplugged version had a physical tableau, a plastic cup and some colored tokens for the execution of the algorithm, the commands were given in pen and paper, and the algorithm was 'executed' by the human instructor. Figure 2 shows students interacting with the game in both versions.

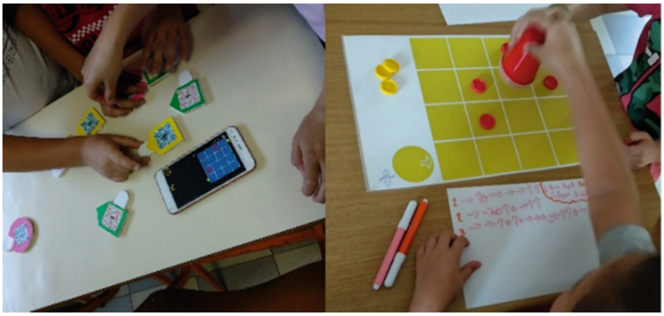

Fig. 2. Photos during students' interaction with ALGO in the AR (left) and unplugged (right) version.

The study adopted a between-subjects approach to avoid the learning effect of the successive use of both conditions.

4.1 Setting and Participants

The study was held during regular school hours in an elementary public school in Syros, Greece as a special activity, in collaboration with their educator, who attended and participated in the process. The role of the educator in the process is critical, whereas there is a turning point from a traditional instructor to a facilitator in modern learning settings. As previous studies have highlighted that educators' understanding and engagement can be limited, in this study we purposefully involved the educator in all sessions. The authors had the role of the instructors during the activity, supported by the educator.

The experiment took place in the context of a classroom. The classroom was divided in two; for half workspaces a smartphone and the physical blocks was laying on the table and for the other half it was the unplugged prototype. Students were sitting around a table in groups of 3 to 4. Then, they were given the instructions by the authors. We described the goal, the commands and the execution process. The experimental group watched a short demonstration, as it was the first time for the student to interact with AR technology.

The participants consisted of a third-grade class of eighteen (18) students gender balanced (9 girls and 9 boys) 8 to 9 years old. As expected, most of them were familiar with mobile devices, such as smartphones and tablets. Although they were novices in computer science concepts, they had some basic prior experience with block-based programming tools.

4.2 Procedure

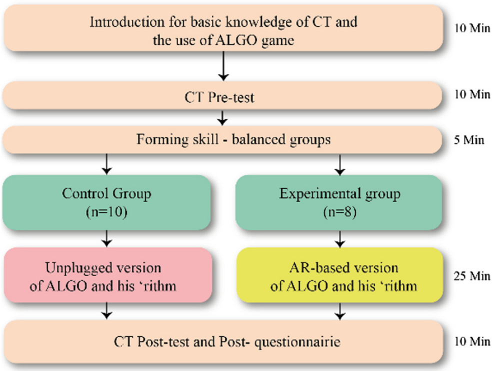

Fig. 3. The study procedure

The process during each session was the following (Fig. 3):

- *Introduction to algorithms and presentation of the activity*: a short introduction to the concepts of CT was held. Then the goal, the commands and the execution process in order to complete the activity was described.
- *CT pre-test*: students had to complete a short pre-test regarding their knowledge on CT concepts.
- *Formation of groups and placement*: students were divided in groups of 3 to 4 in order to work in a collaborative context. There was an existing grouping formed by the educator for every group to be skill-balanced in CS concepts. We respected the existing grouping for the assignment of the experimental and control conditions. Because of the formation of the teams we had two teams of three students and one team of four students (10 in total) in control group and two teams of four students (8 in total) in the experimental group.
- *Game-based learning activity in both conditions (experimental and control)*: the control group interacted with the unplugged prototype, while the experimental group interacted with the AR-based prototype. Each session took place in formal settings in a separated space, during regular school hours. The duration of each

session last around 60 min. During the activity students had to complete three levels i.e. create three programming solutions in both groups.

- *CT post-test and questionnaires*: students had to complete a short post-test and a questionnaire concerning their feelings towards the game-based activity.
- *Wrap up discussion*: a discussion followed the whole process with both students and the educator.

4.3 Measures

The data collection methods adopted during the experiment were observation, think aloud, questionnaires and open discussion, in order to collect both qualitative and quantitative data. The questionnaire aimed to provide feedback about student perception concerning enjoyment and ease of use. Each item of the questionnaire was scaled on the 5 - point Likert scale adapted for children, i.e. using simple language and graphic representations (smileys) instead of numbers. Combined with the questionnaire, we adopted a simplified adaptation of the CT test in order to access the level of CT in students participated in our research, before and after the experiment [12]. Our version of CT test was adapted to the visualization of ALGO's interface (i.e. images of our prototype, game's character etc.) and it focuses on sequences, as a computation concept. At the end of each session an open discussion with the educator was carried out. The performance on programming was evaluated per group, based on the observation of both the research team and the educator. However, the contribution of each student in the collaboration aspect was considered.

5 Results and Discussion

As an educational intervention, the study was quite successful. Both groups participated actively in the process, and most of them managed to solve the challenges. Notably, some of the groups that finished earlier asked for further, more challenging problems, which the instructors provided, at least in the unplugged version.

Regarding the enjoyment factor, the majority of the students of both sessions enjoyed participating in the experiment, as it was shown from the questionnaires where all answers were above neutral (i.e. above 3). The minimum average per question is 3.5, while most of them is around 4.5. The enjoyment was also apparent through observation of students' reactions and comments. For instance, there was a student of the experimental group who asked if they can play it at home. Despite the fact that the AR- based system was an early functioning prototype and engagement factors commonly used in game environments was limited (i.e. only basic animations and sounds was implemented), the new way of interaction provided made the activity challenging and intriguing.

Being the first time for most students to interact with AR technology, it created reactions of surprise and excitement, but also concerns. The most important feedback we received from the students was through the think aloud protocol. We gathered comments like "Wow, it can see the blocks!", referring to the smartphone, but also "Did it get it now?", expressing concern weather the device records the commands, misconceiving the indicators of the action's feedback. Although the students were not familiar with AR technology, which raised concerns and required time to adapt, they were familiar and feeling at ease with the tangible part. Many students described the physical blocks like a puzzle and it was easy for them to understand how they were connected together, which considerably helped to avoid misinterpretations of the semantics - which was crucial as we will explain later below.

Students' prior experience with introductory programming had a mixed effect on the activity. It has been made evident from the CT tests, as within the control group 70% of participants completed correct all answers in pre-test (80% in post-test), while within the experimental group 75% completed correct all answers in pre-test (87% in post-test), and from the discussion with the teacher that they already had an understanding of fundamental concepts through other introductory activities, which made it easier for them to understand what they were supposed to do. However, it also led to some misinterpretations, as they had already created a mental connection of semantics to specific actions. Specifically, in the context of our game the action of collecting an item is successful only if the robot is already on the same block with it. The majority of the students were inclined to use it when ALGO was in a neighboring block. For thirteen (13) students out of eighteen (18), around 68% of all participants, that misconception was obvious in the pre-test. Throughout the activity most of the students of the experimental group understood the difference of semantics through a trial and error procedure. They took the initiative to try out the meaning of the blocks/commands, before they start working towards the goals. On the other end, the control group did not have that opportunity, as they had to wait for the instructor to check their solution. In this respect, the tangible AR activity provides the opportunity for more initiative actions, personalized features and direct feedback, while the unplugged version is solely dependent on the instructor for feedback.

Concerning collaboration, we observed different types of organization and management of the tasks and roles inside every group. We did not give any directions regarding collaboration patterns and there was no specified role assignment inside the group, i.e. someone to control the device or someone else to issue the commands. Therefore, every group created their own pattern of collaboration. For instance, in the experimental group one team decided that one member would solely handle the device and the other decided to take turns. The AR-based version facilitated more structured and clear collaboration patterns, as it has clear sequence of actions guided by the system. On the other end, the unplugged version does not define clear steps by itself, as they face all at once the "execution tableau", the problem and the means to create the solution, which created distractions and made the role of educator crucial. Indeed, the most incidents of disagreement occurred during the control group session, which considerably affected the total time spend on every goal, prolonging the session. That indicates two things, the AR-based version (a) facilitates control by the instructors, which means that during the experimental session, instructors could maintain control and guidance easier as they had less involvement in the activity itself and (b) supports more effective and structured ways of collaboration.

Finally, through our open discussion with the educator another challenge occurred for long term use of the AR-based version of ALGO in formal settings, concerning the need of educators of keeping track of students' performance. The game is keeping the score temporarily but it does not provide the educator with the feedback they require in long term. Therefore, they have to keep track for themselves, which increases the level of involvement. During this study there was some limitations in order to provide more concrete evidences, such as our early functioning prototype i.e. only a few levels were implemented, while more complex computational concepts are planned for future iterations. It is of significant importance to test our approach with a full functioning prototype of ALGO and his 'rithm in long term use, although open discussion with the educator gave us important feedback concerning that matter.

6 Conclusions

In this paper we present the results from a preliminary study in an elementary school that compared a tangible AR game with the same game in an unplugged version in teaching computational thinking, to examine the effect of the interface on student motivation, effectiveness and teaching practice. The results were encouraging, in terms of improving engagement and collaboration, while they provided interesting insights towards teaching practice. Both approaches show positive outcomes on students' enjoyment. The AR-based version seems to prevail in terms of (a) collaboration, (b) educator's control, (c) excitement; emerge from interacting with technology combined with physical object, (d) direct feedback, while both approaches share (d) the benefit of low cost or widely available equipment.

Field evaluations are very meaningful, as real context requirements and limitations surface. More field sessions focusing on qualitative and quantitative results concerning educational goals in long term use of our approach are needed. Some interesting directions for this approach arose through the study. It will be interesting to create a more challenging context, outside the game, i.e. by expanding the interaction between the groups in term of collaboration or competition in kind of a multiplayer game.

A section of free play is a plausible solution for future work and research, in order for students to try out and understand the meaning of semantics before they start playing, without being discouraged, as a trial and error way tend to make children feel disenchanted. Finally, as an educational tool intended for classroom settings, an educator version that will keep track of students' performance, in long term, may be an interesting direction.

References

1. Walker, C.O., Greene, B.A., Mansell, R.A.: Identification with academics, intrinsic/extrinsic motivation, and self-efficacy as predictors of cognitive engagement. Learn. Individ. Differ. **16**(1), 1–12 (2006)
2. Papastergiou, M.: Digital game-based learning in high school computer science education: impact on educational effectiveness and student motivation. Comput. Educ. **52**(1), 1–12 (2009)
3. Guzdial, M.: Education Paving the way for computational thinking. Commun. ACM **51**(8), 25–27 (2008)

4. Myers, B.A.: Taxonomies of visual programming and program visualization. J. Vis. Lang. Comput. **1**(1), 97–123 (1990)
5. Bell, T., Alexander, J., Freeman, I., Grimley, M.: Computer science unplugged: school students doing real computing without computers. N. Z. J. Appl. Comput. Inf. Technol. **13** (1), 20–29 (2009)
6. McNerney, T.S.: From turtles to tangible programming bricks: explorations in physical language design. Pers. Ubiquitous Comput. **8**(5), 326–337 (2004)
7. Goyal, S., Vijay, R.S., Monga, C., Kalita, P.: Code Bits: an inexpensive tangible computational thinking toolkit For K-12 curriculum. In: Proceedings of the TEI 2016: Tenth International Conference on Tangible, Embedded, and Embodied Interaction, pp. 441–447. ACM (2016)
8. Klopfenstein, L., Fedosyeyev, A., Bogliolo, A.: Bringing an unplugged coding card game to augmented reality. INTED Proceedings, pp. 9800–9805 (2017)
9. Kranov, A. A., Bryant, R., Orr, G., Wallace, S. A., Zhang, M.: Developing a community definition and teaching modules for computational thinking: accomplishments and challenges (2010)
10. Kazimoglu, C., Kiernan, M., Bacon, L., MacKinnon, L.: Learning programming at the computational thinking level via digital game-play. Procedia Comput. Sci. **9**, 522–531 (2012)
11. Lee, I., et al.: (2011). Computational thinking for youth in practice. ACM Inroads **2**(1), 32–37
12. Brackmann, C.P., et al: Development of computational thinking skills through unplugged activities in primary school. In: Proceedings of the 12th Workshop on Primary and Secondary Computing Education- WiPSCE 2017, pp. 65–72. ACM Press, New York, New York, USA (2017)
13. Horn, M.S., Jacob, R.J.K.: Designing tangible programming languages for classroom use. In: Proceedings of the 1st International Conference on Tangible and Embedded Interaction (TEI 2007), (February), pp. 159–162 (2007)
14. Paulo Blikstein (Stanford University, USA), Arnan Sipitakiat (Chiang Mai University, Thailand), Jayme Goldstein (Google), João Wilbert (Google), Maggie Johnson (Google), S. V. (Google), & Zebedee Pedersen (Google), W. C. (IDEO). (2016). Project Bloks : designing a development platform for tangible programming for children
15. Carmigniani, J., Furht, B.: Augmented reality: an overview. In: Handbook of Augmented Reality, pp. 3–46. Springer, New York (2011)
16. Teng, C.-H., Chen, J.-Y., Chen, Z.-H.: Impact of augmented reality on programming language learning. J. Educ. Comput. Res., 073563311770610 (2017)
17. Santos, M.E.C., Chen, A., Taketomi, T., Yamamoto, G., Miyazaki, J., Kato, H.: Augmented reality learning experiences: survey of prototype design and evaluation. IEEE Trans. Learn. Technol. **7**(1), 38–56 (2014)
18. Azuma, R.T.: A survey of augmented reality. Presence Teleoperators Virtual Environ. **6**(4), 355–385 (1997)
19. Sapounidis, T., Demetriadis, S., Stamelos, I.: Evaluating children performance with graphical and tangible robot programming tools. Personal Ubiquitous Comput. **19**(1), 225–237 (2015)
20. Sung, H.Y., Hwang, G.J.: A collaborative game-based learning approach to improving students' learning performance in science courses. Comput. Educ. **63**, 43–51 (2013)