Sustaining Cultures of Participation by Fostering Computational Thinking Skills through Game-Play

Daniela Fogli, Federico Danesi
University of Brescia
Brescia, Italy
daniela.fogli@unibs.it

Alessio Malizia, Tommaso Turchi, David Bell
Brunel University
London, UK
{alessio.malizia, tommaso.turchi, david.bell}@brunel.ac.uk

ABSTRACT
The adoption of a meta-design approach to system development opens up opportunities for transforming consumer cultures to cultures of participation. To this end, meta-design must create the conditions for such participation by supporting end users to appropriate the design skills necessary for system evolution, especially those related to Computational Thinking (CT), in new and engaging modalities. In this paper, we propose a novel approach to fostering CT skills that combines Game-Play learning with Tangible User Interfaces and Virtual Reality (VR). In the resulting system, called TAPASPlay, two players act as alchemists forging swords and shields to fight each other. They build them through a puzzle-based interaction with a tabletop interface, using smartphones as tangible objects. Finally, the players can enjoy the battle in VR using Google Cardboards. In this way, players can develop analysis, abstraction and problem solving abilities, i.e. suitable CT skills for meta-design and supporting cultures of participation.

Author Keywords
End-user development; meta-design; computational thinking; game-based learning; tangible interaction.

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

User-centered design and participatory design are usually advocated as successful approaches to designing systems that properly fit with users' work practices, preferences and needs. Both approaches foresee user involvement at design time to inform designers about system functional and non-functional requirements. Such an involvement may be regarded as a one-way communication (from users to designers) in case of user-centered design, where users are for example observed or interviewed; or may become a two-way communication in participatory design, where users are given a voice and can actively participate with their ideas in design decisions. In addition, nowadays there is a growing request of hardware and software systems that can be easily tailored and extended by end users at the use time. This is not only true for traditional information systems [7] or spreadsheet-based applications [2], but also personal devices [6][11] and environments (the so-called "smart home") [3][5].

Meta-design has been proposed as a novel approach to designing open systems that can progressively evolve in the hands of end users, by means of end-user development (EUD) methods and techniques [10]. In this way, meta-design aims at sustaining a cultural transformation, by supporting end users to become co-designers and end-user developers [9]. On the one hand, such a progressive transformation from consumer cultures to cultures of participation [8] is facilitated by current technological innovations, from Web 2.0, to the Internet of Things, to tangible interactive spaces [1][21]; on the other hand, however, not all end users are ready today for such a transformation or willing to acquire the new skills necessary for an effective participation. To this end, meta-design is also concerned with the creation of the social conditions for end-user participation at design and use time, by sustaining end users to appropriate the design culture and the technical notions necessary for system evolution [4]. To create such social conditions, meta-design should transfer to the end users those Computational Thinking (CT) skills [22] that can be useful to sustain cultures of participation.

CT skills are typical of programmers and software designers and, even though a unique definition does not yet exist, we can sum up most of literature attempts to define CT skills in a set comprising *abstraction*, *algorithmic thinking*, *decomposition*, and *problem solving*. Mastering those skills lowers the learning barrier when approaching a programming activity. For this reason, traditional approaches to teaching CT skills involve visual programming languages, such as Scratch [18], or game design activities, such as AgentSheets [16], properly oriented to teach the concepts underlying imperative programming (symbolic representation, conditionals, loops, operators, etc.). These "making" activities [14] encourage to cohesively combine multiple ideas into an organized

GHITALY17: 1st Workshop on Games-Human Interaction, April 18th, 2017, Cagliari, Italy.

process to produce an artifact that solves problems. Indeed, CT skills go beyond programming constructs ("conceptualizing, not programming" [22]): in her original definition, Wing assumes a higher-level perspective, arguing that CT skills create a new mindset oriented towards problem solving, thanks to the ability of thinking at different levels of abstraction and at decomposing problems into sub-problems [22]. This can bring several benefits in everyday life, including, in our opinion, a more effective participation by end users in the design and development of their systems both at design time and use time.

Starting from Wing's ideas, Repenning and colleagues modeled computational thinking as an iterative process structured in three stages [17]: 1) *problem formulation*, namely a verbal or diagrammatic conceptualization of the problem, where abstraction plays a fundamental role; 2) *solution expression*, when the solution is formulated in a way that can be understood by a computer (the most common, but not unique, tool used in this stage is programming); 3) *execution and evaluation,* i.e. those activities allowing one to visualize and assess the outcome of the other two stages.

In this paper, we describe TAPASPlay, an extension of TAPAS (TAngible Programmable Augmented Surface) [20][21], specifically designed to foster CT skills through a game-based learning approach, by favoring the CT iterative process described in [17]. TAPASPlay is based on the alchemist' metaphor: two players compete to be the best alchemist and, by applying transformations on metals, they forge swords and shields to fight each other. Like TAPAS, TAPASPlay adopts a puzzle-based interaction with a tabletop surface, where a smartphone plays the role of tangible object used for dragging and dropping digital objects that will make up the swords. The final battle is eventually enjoyed in Virtual Reality (VR) by wearing Google cardboards.

TAPASPlay has been designed to help bridging the gap between the end user and the designer roles. Ideally, in the meta-design discourse, the user would be able to grasp different aspects of the system (from features, to standards, to usability issues) and actively contribute to the design itself. Unfortunately, there are some language and conceptual barriers that prevent the end-user communication with technologists (i.e. software engineers). Understanding an algorithmic solution to a problem and thus being able to participate in the selection of the right solution by helping modeling the problem is a very relevant activity in a meta-design approach, but in our opinion requires exactly those CT skills mentioned above.

Furthermore, our everyday life depends more and more on algorithms [19]: think about how many times a day we interact with algorithms, from automatic checkouts in supermarkets and e-banking to booking a flight online (and during that same flight as well: in fact, 95% of the whole journey is flown by an autopilot running an algorithm). By acquiring CT skills the end user would be able to understand, participate and trust algorithmic solutions and thus play a much relevant active role in the meta-design discourse.

The paper is organized as follows: the following section contextualizes and compares the approach presented here with recent literature work; then we describe TAPASPlay and the phases of the gameplay; finally, the last section discusses the main features of the system and draws some conclusions.

GAME-BASED LEARNING

Digital games proved attractive and engaging for all groups of people and therefore, Game-Based Learning (GBL) has been proposed as one pedagogical framework for developing CT skills [12]. In order to help acquiring CT skills two main approaches have been introduced in GBL: learning through designing games and learning through game-play. Whilst learning by designing games has been studied quite extensively (e.g. Scratch, Alice and AgentSheets), the game-play approach is relatively new, especially in light of employing it to improve CT skills.

As an example, *Program Your Robot* [13] is a recent game prototype developed to support players in practicing the five core skills that the authors identified as fundamental for computational thinking, namely *problem solving*, *building algorithms*, *debugging*, *simulation* and *socializing*. It is a puzzle solving game in which the player has to assist a robot to reach a certain point on a grid. The robot will follow very simple instructions given in the form of an algorithm, while the score depends on conditions, for example if two functions have been declared before being called in the algorithm. It differs from the software applications for game design mentioned before, since those ones can be deemed programming languages to all effects, while *Program Your Robot* is conceived as a serious game. But above all, tools like Scratch were designed in order to teach the basics of programming and to show how fun it can be. Instead, Kazimoglu and his colleagues [13] were moved by the goal of creating a game that could explicitly foster CT skills.

CTArcade [15] is another serious game designed with the target of boosting computational thinking in players by letting them formalize their tacit knowledge and make a step towards abstraction. In CTArcade users have to implement a set of rules that are observed by a character while playing Tic-Tac-Toe. Making these rules explicit is considered a very important process, because people often apply them in a natural, perhaps unconscious way and normally there is neither occasion nor reason to transform this knowledge into abstract instructions.

These systems use a traditional interaction style based on keyboard and mouse; on the contrary, even though TAPASPlay shares with them the same goal, that is fostering CT skills through a game-play approach, it leverages on an interaction style that relies on the use of tangible objects and virtual reality. TAPASPlay fits also within the realm of Constructionist Video Games [25], namely designed computational environments in which players construct personally meaningful artifacts to overcome artificial conflicts or obstacles resulting in quantifiable outcomes.

TAPASPLAY

The novelty of TAPASPlay is to combine game-play with tangible user interfaces and Virtual Reality to teach CT skills. The game is intended for an audience with little or no experience in programming, which is trained in such computational abilities to become able to participate in system design and end-user development activities.

TAPASPlay has been developed starting from TAPAS [21], an End-User Programming (EUP) platform for pervasive display repurposing in the wild. Therefore, as for TAPAS the interaction with TAPASPlay requires a pervasive display or a tabletop surface, an RGB camera and a smartphone. The smartphone is used both as a computing device and as a tangible object, and its movements on the display or surface are tracked by the RGB camera that locates the position of a fiducial marker displayed on the phone screen and uses it as reference point. TAPASPlay has been implemented as a web application that is projected on the pervasive display or tabletop surface and is able to interact with the players' smartphones. Differently from TAPAS, TAPASPlay can be regarded as a constructionist video game aimed at satisfying the following requirements:

- It must provide both an entertaining and an educational experience. The latter has the goal of fostering Computational Thinking skills, while the use of Virtual Reality should boost the players' engagement.
- The game must feature a metaphor suitable to a VR representation, which can be visualized by wearing affordable goggles (e.g. Cardboards by Google).
- The interaction with the game should be based on a puzzle metaphor, like the original TAPAS system. This means that TAPASPlay has to communicate the existence of constraints and to support the gameplay through puzzle pieces and their shapes, aiding users whilst giving constraints in their selection process.

TAPASPlay is thus a game to be played in a player versus player modality. Players compete to be the best alchemist forging three swords and three shields, made of three different metals. The game features three phases:

1. defining the offensive strategies, by means of forging swords;
2. defining the defensive strategies, by means of forging shields;
3. visualizing the representation of a battle in a VR headset.

Forging swords

During the first phase, each half of the tabletop screen is available for a player to define three offensive strategies, which will be visualized as three swords. In order to accomplish that, players have to attach transformations, represented as pieces of a puzzle, to a halo surrounding the smartphone of the user on the main display. The halo, with its three hilts, follows the movement of the dragged smartphone and, when a collision with a puzzle piece is detected, such piece is attached to the vertically oriented hilt under some given conditions. The three swords are defined one at a time. For example, in Figure 1, each player is creating his/her first sword.

Figure 1. Forging swords through tangible and puzzle-like interaction.

Each strategy is a sequence of transformations taken from a randomly generated set shown on the main display (Fig. 2).

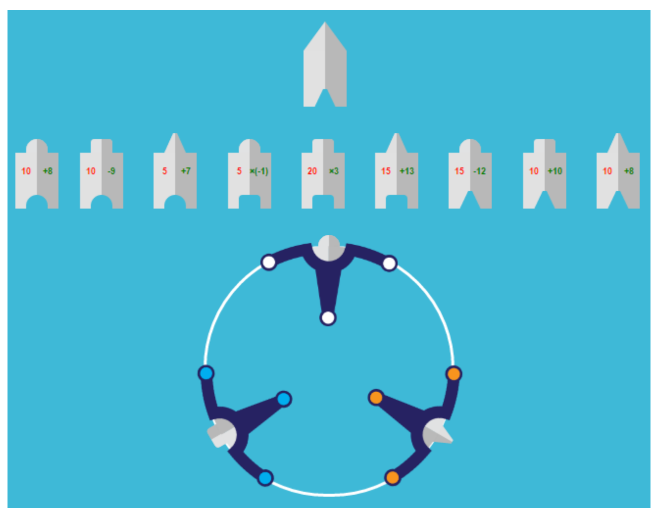
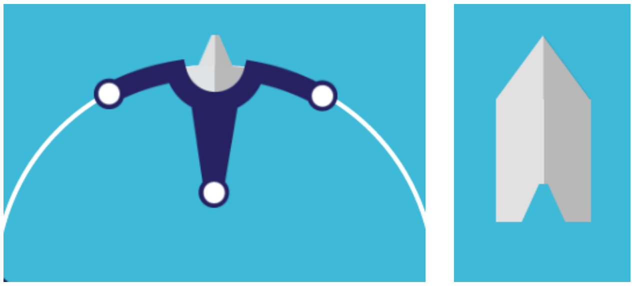


Figure 2. Defining an offensive strategy: the set of transformations is displayed, as well as the main halo with three hilts and the final piece.

A hilt attached to the main halo surrounding the player's smartphone represents the start of a sequence (Fig. 3(a)), while the final piece has a shape that resembles the tip of a sword (Fig. 3(b)).



(a) (b)

Figure 3. (a) An example of initial state of the sword, (b) an example of final piece.

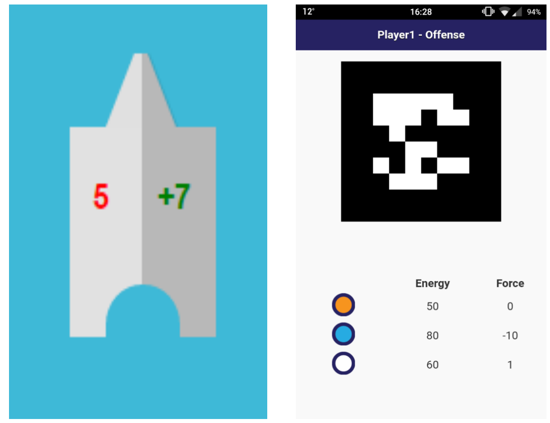
Each sword is made of a different type of metal, determined by the shape of the final puzzle piece (e.g., in Fig. 3(b) the shape of the final piece is triangular). Every puzzle piece has an input and an output shape. There are three shapes in total, *round*, *square* and *triangular*, which in turn correspond to three types of metal, namely *bronze*, *iron* and *steel*. So, if a puzzle piece has a round input shape and a triangular output shape as in Fig. 4(a), it is equivalent to a transformation that turns bronze into steel.

The aim of this first phase is to maximize the *force points* of each sword, which can be earned by attaching transformations to the sequence. However, every transformation consumes an amount of *energy points*. More precisely, a transformation is a tuple of four values: 1) an input shape, 2) an output shape, 3) an amount of energy points, displayed on the transformation (left half in Fig. 4(a)), and 4) the force points gained, displayed on the transformation as well (right half in Fig. 4(a)).

In order to apply a transformation at a certain stage of the strategy, two conditions need to be fulfilled: 1) the input shape of the transformation is the same as the output shape of the last transformation attached to the sequence (or, if the transformation applied is the first of the sequence, the input shape has to be the same as the output shape of the initial state); 2) the alchemist must have an amount of energy points greater or equal than the one showed on the transformation. Once a transformation is applied (supported by a "magnetic effect" on the puzzle piece provided by the system), the energy points of the alchemist are decreased by the energy points of the transformation, while the force points of the strategy can be increased, decreased or multiplied, depending on the operation suggested by the transformation.

Players can see a feedback of their operation on their smartphone, since force and energy points presented on their screen are updated according to the values displayed on the transformation. See for example Fig. 4(b), where the correspondence between swords and values displayed on the smartphone is given by the cue balls matching the gems of the hilts showed on the halo.

The initial state of each sword consists of an output shape attached to a hilt on the halo, an amount of force points, and an amount of energy points. The final state is reached when the player is satisfied with its sequence of transformations and decides to – and can – attach the final piece to the sword. This is a special transformation that does not modify force nor energy points, but only suggests the final constraint on the sequence - Fig. 3(b).



(a) (b))

Figure 4. (a) An example transformation, (b) the energy and force points shown on the smartphone.

Forging shields

The defensive strategy consists of allocating an amount of defense points into three shields, each one corresponding to a different metal just like the swords, through an interface displayed on the smartphone. The choice should be based on a couple of considerations: how the player guesses the opponent distributed force points on the different swords and the transformations chosen for building her/his own swords. For instance, if a player struggled to compose the strategy for the bronze sword, then he/she might consider allocating most of defense points into the bronze shield, in order to counterpoise her/his weak offensive strategy.

Enjoying the battle in VR

When both the previous game phases terminate, an Android application showing the resulting Virtual Reality video becomes available from the server. By receiving the score of the game from the web application, the server provides each player with a different video to be played. For instance, if Player 1, who used the halo with blue hilts, managed to reach the highest score, the video will show a knight wearing a blue armor defeating the opponent dressed in red; otherwise a video with reversed roles will be played. In order to correctly visualize the content of the app, both players are asked to wear goggles as Google Cardboard. The VR video shows two knights armed with sword and shield. At the beginning, a button containing the text “Start” needs to be selected in order to play the animation. A pointer placed at the center of the user’s sight suggests that, to push the button, it is required to gaze at it. After having pressed the button, the two knights approach the center of the scene and, when they are close enough, they start dueling. They exchange a few hits for a little while, then the knight on the left takes a few steps back, runs toward the opponent and launches the decisive blow. The wounded knight falls on the ground and, while the winner cheers, a text appears on the background, confirming which player won (Fig. 5).

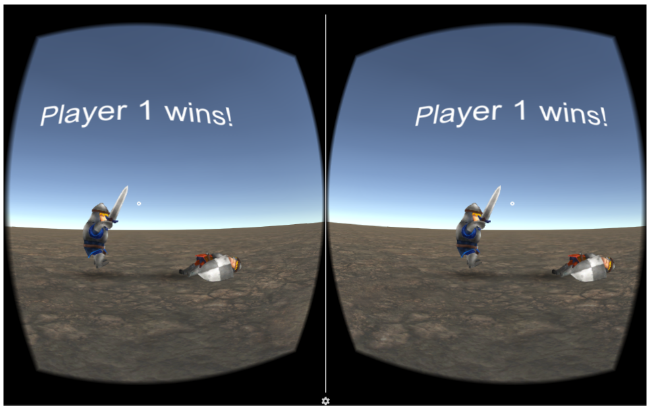


Figure 5. Visualizing the outome of the battle in VR: the duel has ended with the victory of Player 1.

DISCUSSION AND CONCLUSION

The growing interest in Computational Thinking is also witnessed by very recent literature [23], which describes how CT is becoming more and more important in student and teacher education. In this paper, we suggest that CT skills are fundamental to sustain cultures of participation and allow end users to collaborate to system design and evolution at use time. For this reason, contrarily to other block-based approaches, in TAPASPlay blocks do not represent programming statements (like for example, the "if-then" block in Scratch) but remain at a higher level of abstraction, to promote problem decomposition abilities rather then programming ones.

Like TAPAS, TAPASPlay considers tangible user interfaces and physical object manipulation as fundamental tools to make user activities more engaging. Indeed, in a study with children aged 5-9, it has been demonstrated that tangible programming has the potential to help children cultivate skills such as abstraction and problem decomposition [24]. Similarly, we would like to demonstrate in the future that end users can more easily acquire CT skills through tangible interaction, and thus become proficient in end-user development activities. In addition, TAPASPlay includes Game-Based Learning to make the experience engaging and social. In particular, we would like to contribute to the recent research trend that explores learning through game play [13], instead of learning through designing systems, as a new pedagogical approach to fostering CT skills.

In TAPAS a challenge was observed concerning the duality of composing and executing workflows, both requiring the use of tangible interaction through smartphone assuming two different meanings, as tangible object and as source of data [21]. On the contrary, TAPASPlay detaches composition from execution by offering two different interaction styles and tools: puzzle-based interaction with a display/surface where a smartphone is used for composing the strategy (problem solving); whilst, VR is adopted for checking solution execution. This mechanism fosters the design-debug-run stages, three key aspects of Computational Thinking [13], or in other terms, the process of problem formulation-solution expression-execution and evaluation [17].

Analysis, abstraction, decomposition, and automation all come into this game-play. While automation is supported by VR, analysis, abstraction and problem decomposition are types of reasoning that players are supposed to apply when trying to maximize the force points, under the constraints represented by shapes and limited energy points. As a matter of fact, the choice of displaying all transformations together at the start of the game makes deliberately complex for the player to formulate a straightforward solution. On the other hand, if the player is “lazy” and does not want to apply a methodic decomposition process, but merely tries to satisfy the constraints, a solution would be reached, but chances that it is a good one are quite low, in terms of force points. Therefore, the player would try to “fix it” by analyzing it and identifying the weakest subsequence of transformations. Hence, the solution would be reformulated by replacing the poor part with a different sequence of pieces. This process might be repeated several times, inducing the player to iteratively apply the model of computational thinking process proposed in [17].

Let us notice that all the above skills are indeed crucial for the end users to play an active role in the algorithmic solution proposed and discussed with technologists, therefore ultimately unveiling the end users’ inner model of the problem scenario tackled by the meta-design approach. Lastly, Kazimoglu et al. [13] add also socialization to CT skills fostered by learning through game play. In TAPASPlay, it is reasonable to expect that the gaming experience could lead users to socialize by sharing thoughts about their approaches, thus stimulating cooperative strategy development useful in co-design processes.

TAPASPlay is however a first proposal to fostering CT skills in end users. Experiments with domain experts and industrial designers will be carried out in the next future to demonstrate the validity of the idea. Furthermore, several extensions of TAPASPlay have been already planned, in order to tailor the system to end users' characteristics and introduce different levels of complexity in the game. At the moment, only a VR simulation of the battle is available as outcome of the game; however, the system could be extended adding a more interactive functionality that better resembles the debugging activity, in which players can compare step-by-step how they built their swords and eventually see what was the optimal solution.

ACKNOWLEDGMENTS

We thank the University of Brescia and Brunel University London for supporting us in this collaboration with grant “Bando Tesi all’estero 2016-2017” DR-291-2016.

REFERENCES

1. E. G. Arias, H. Eden, G. Fischer. The Envisionment and Discovery Collaboratory (EDC): Explorations in Human-Centered Informatics. San Rafael, California: Morgan & Claypool Publishers, 2016.
2. M. Burnett. What is End-User Software Engineering and Why Does It Matter? In: V. Pipek, M. B. Rossen, B. deRuyter & V. Wulf (Eds.), End-User Development (pp. 15-28). Heidelberg: Springer, 2009.
3. F. Cabitza, D. Fogli, R. Lanzilotti, A. Piccinno. Rule-based Tools for the Configuration of Ambient Intelligence Systems: a Comparative User Study. Multimedia Tools And Applications, 76(4), pp. 5221–5241, 2017.
4. F. Cabitza, D. Fogli, A. Piccino. Fostering participation and co-evolution in sentient multimedia systems. Journal of Visual Languages and Computing, 25(6), pp. 684-694, 2014.
5. J. Coutaz, J. L. Crowley. A First-Person Experience with End-User Development for Smart Homes. IEEE Pervasive Computing, April-June, pp. 26-39, 2016.
6. J. Donado, F. Paternò. Puzzle: A mobile application development environment using a jigsaw metaphor. Journal of Visual Languages and Computing, 25(4), pp. 297-315, 2014.
7. C. Dörner, J. Heß, V. Pipek. Improving Information Systems by End User Development: A Case Study. In Proc. European Conference on Information Systems (ECIS), St. Gallen, Switzerland, pp. 783-794, 2007.
8. G. Fischer. Understanding, fostering, and supporting cultures of participation. Interactions, 18(3), pp. 42-53, 2012.
9. G. Fischer, D. Fogli, A. Piccinno. Revisiting and Broadening the Meta-Design Framework for End-User Development. In: F. Paternò and V. Wulf (Eds.), New Perspectives in End-User Development, Springer, in press.
10. G. Fischer, E. Giaccardi. Meta-Design: A Framework for the Future of End User Development. In H. Lieberman, F. Paternò, V. Wulf (Eds.), End User Development (Vol. 9, pp. 427-457). Dordrecht, The Netherlands: Springer, 2006.
11. R. Francese, M. Risi, G. Tortora, M. Tucci. Visual Mobile Computing for Mobile End-Users. IEEE Transactions on Mobile Computing, 15(4), pp. 1033-1046, 2016.
12. D. Holbert, N. R. Horn, M. S. Wilensky. Computational Thinking in Constructionist Video Games. International Journal of Game-Based Learning, 6(1), pp. 1-17, 2016.
13. C. Kazimoglu, M. Kiernan, L. Bacon, L. MacKinnon,. Learning Programming at the Computational Thinking Level via Digital Game-Play. Procedia Computer Science, 9, pp. 522-531, 2012.
14. D. Kotsopoulos, L. Floyd, S. Khan, I.K. Namukasa, S. Somanath, J. Weber, C. Yiu. A Pedagogical Framework for Computational Thinking, Digital Experiences in Mathematics Education, pp.1-18, 2017.
15. T. Y. Lee, M. L. Mauriello, J. Ahn, B. B. Bederson, CTArcade: Computational thinking with games in school age children, Int. Journal of Child-Computer Interaction 2, pp. 26-33, 2014.
16. A. Repenning. AgentSheets®: an Interactive Simulation Environment with End-User Programmable Agents. Interaction 2000, Tokyo, Japan, 2000.
17. A. Repenning, A. Basawapatna and N. Escherle. Computational Thinking Tools. In Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2016), pp. 218-222, 2016.
18. M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman, Y. Kafai. Scratch: programming for all. Communications of the ACM, 52(11), pp. 60-67, 2009.
19. B. Shneiderman, C. Plaisant, M. Cohen, S. Jacobs, N. Elmqvist, N. Diakopoulos. Grand challenges for HCI researchers. Interactions, pp. 24-25, 2016.
20. T. Turchi, A. Malizia. Fostering computational thinking skills with a tangible blocks programming environment. In Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2016), pp. 232-233, 2016.
21. T. Turchi, A. Malizia, A. Dix. TAPAS: A tangible End-User Development tool supporting the repurposing of Pervasive Displays. Journal of Visual Languages and Computing, 2016, 39, pp. 66-77, 2016.
22. J. M. Wing. Computational thinking. Communications of the ACM, 49(2), pp. 33-35, 2006.
23. A. Yadav, C. Stepheson, H. Hong. Computational Thinking for Teacher Education. Communications of the ACM, 60(4), pp. 55-62, 2017.
24. D. Wang, T. Wang, Z. Liu. A Tangible Programming Tool for Children to Cultivate Computational Thinking. The Scientific World Journal, vol. 2014, Article ID 428080, 2014.
25. D. Weintrop, N. Holbert, M.S. Horn, U. Wilensky. Computational thinking in constructionist video games. International Journal of Game-Based Learning (IJGBL), 6(1), pp.1-17, 2016.