Multimedia Tools and Applications (2019) 78:13649–13673
https://doi.org/10.1007/s11042-019-7229-9

Fostering computational thinking through collaborative game-based learning

Tommaso Turchi[1] · **Daniela Fogli**[2] · **Alessio Malizia**[3,4]

Received: 10 May 2018 / Revised: 21 December 2018 / Accepted: 15 January 2019 /
Published online: 4 February 2019

Abstract
Algorithms are more and more pervading our everyday life: from automatic checkouts in supermarkets and e-banking to booking a flight online. Understanding an algorithmic solution to a problem is a very relevant activity to improve end-users' involvement. To this end, adopting a meta-design approach may help to support end-users to appropriate the design skills necessary for contributing to system design, in new and engaging modalities. By acquiring Computational Thinking (CT) skills (e.g., algorithmic thinking, abstraction), end-users will be able to understand and trust algorithms, while at the same time participate in the design and development of systems evolving in accordance with their needs. In this work, we focus on two different ways of improving CT skills: playfulness and collaboration. We introduce a game-based system, TAPASPlay, to foster CT skills and we report the results of an exploratory study with 18 users; our hypothesis is that learning CT through gameplay is effective and we tested it by involving participants in game sessions providing playful experience and collaborative learning.

Keywords Computational thinking · Gameplay · Tangible user interface · Constructionist video games

This is an extended and revised version of a paper that was presented at the 2017 Workshop on Games-Human Interaction [19]. This paper significantly expands over the presentation and experimental validation of TAPASPlay.

✉ Tommaso Turchi
tommaso.turchi@brunel.ac.uk

Daniela Fogli
daniela.fogli@unibs.it

Alessio Malizia
a.malizia@herts.ac.uk

[1] Department of Computer Science, Brunel University London, Uxbridge, UK

[2] Department of Information Engineering, University of Brescia, Brescia, Italy

[3] School of Creative Arts, University of Hertfordshire, Hatfield, Hertfordshire, UK

[4] Faculty of Logistics, Molde University College, Molde, Norway

1 Introduction

Nowadays, society is highly dependent on algorithms [52]: from managing our investments and daily life schedule to aiding cancer diagnosis. Being able to understand and trust algorithmic solutions, and to participate in the design and development of such solutions can bring several benefits in everyday life, making everyone able to succeed in today's complex and technological society [7].

This can be achieved by creating the socio-technical conditions for empowering users, as problem owners, to participate in system evolution [17]. In particular, End-User Development (EUD) methods [34] usually contribute to fostering the technical conditions for such participation. They are useful not only in traditional information systems [15] or spreadsheet-based applications [8], but also for tailoring personal devices [14, 22] or smart environments [10, 13]. Furthermore, it has been recently demonstrated [51] how workers can benefit from EUD also from the economic point of view. Scaffidi's work shows that American workers able to use spreadsheets or databases and perform some programming tasks earned in the order of 10% more than peers who did neither of those activities.

However, not all end-users are ready or willing to acquire the new skills necessary for an effective participation in system development. To this end, a meta-design approach [17, 18] should aim at creating the social conditions for EUD [9], by supporting end-users to appropriate those Computational Thinking (CT) skills [64] necessary for understanding and contributing to system evolution.

Ideally, in the meta-design discourse, the user would be able to grasp different aspects of the system (from features and standards to usability issues) and actively contribute to the design itself. Understanding an algorithmic solution to a problem and thus being able to participate in the selection of the right solution and helping to model it are very relevant activities in a meta-design approach. In addition, there are often language and conceptual hurdles that create a communication gap between end-users and software designers. Acquiring CT skills may help end-users overcome those hurdles [53].

CT is a recent notion, which was introduced by Seymour Papert in [43], before Jeannette Wing discussed it widely for the first time in her seminal work [64]. Since then, many scholars have tried to define it and find a solid way to assess the set of skills that make it up. According to Wing and other research scholars, CT is both a thought process and a skill set. More precisely, it is regarded as the thought process involved in formulating problems and their solutions so that the "solutions are represented in a form that can be effectively carried out by an information-processing agent" [65]. So, basically, Wing argues that CT is mostly about problem-solving, as well as the capability of using abstraction, problem decomposition, and algorithmic thinking, which are also regarded as highly relevant CT skills [41, 59, 65].

Therefore, CT skills should become an important piece of the end-users' background, like reading, writing and calculating. In line with this consideration, several initiatives all over the world are aimed at ensuring that students develop CT skills at K-12 level [33, 59, 67].

Since CT skills are usually related to programming, most of the approaches to teaching such skills involve visual programming languages (e.g., [48]) or game design activities (e.g., [45]) aimed at teaching the concepts underlying imperative programming (e.g., symbolic representation, conditionals, loops, operators). However, we agree with Wing that CT goes beyond programming since it fosters the creation of a new mindset oriented towards problem-solving, thanks to the ability to think at different levels of abstraction and at

decomposing problems into sub-problems. In other words, CT should focus on "conceptualizing" rather than "programming" [64].

In this work, we focus on a game-based approach to foster CT skills, and in particular, our research question is "Is gameplay an effective way to foster learning of CT skills?". More precisely, we focused on two aspects of our research question:

1. Can we provide a playful way of learning CT skills?
2. Can collaborative learning help to improve CT skills?

Our hypothesis is that through the design of an appropriate gameplay system and involvement of Tangible User Interfaces (TUIs), a good level of playfulness and support at learning CT can be achieved. Furthermore, we think that using a digital shared surface with tangibles can foster collaborative learning to further improve CT skills.

We tested our hypothesis by involving 18 UK secondary school Key Stage 4 (15 years old) female students in collaborative gaming sessions with a prototype of a game we developed called TAPASPlay. The participants had no previous programming experience and only a small group had some previous experience on block-oriented programming.

2 Related works

2.1 Computational thinking: definitions and approaches

As highlighted in the recent review by Shute et al. [53], there is little agreement on the exact definition of CT and over the years several different definitions have been proposed. Anyway, most of the literature works attempt to define CT skills as a multifaceted skillset comprising abstraction, algorithmic thinking, problem decomposition, and debugging. Shute et al. [53] add iteration and generalization as two more skills that are important in CT. *Abstraction* is the ability to think at different levels, and the capability of modelling the core aspects of problems/systems by discarding irrelevant details [64]. *Algorithmic thinking* refers to the ability to create procedures as ordered steps to implement solutions [1]. *Problem decomposition* is concerned with breaking a problem down into manageable units [64]. *Debugging* is particularly emphasized in [4] as the ability to identify and fixing errors when algorithms do not provide the expected solution. According to [53], *iteration* is important for repeating design processes to refine solutions and *generalization* is fundamental to transfer CT skills to a wide variety of contexts.

Since CT skills are related to computer programming, there are many emerging initiatives such as the Hour of Code[1] and most of the literature focuses on methods and tools for teaching computer programming [38, 48]. However, as also suggested by the above facets and by Wing's seminal work, CT does not coincide with programming; rather, it "includes a way of thinking about everyday activities and problems" [53]. Such problems are often ill-structured (or wicked), in that they may have neither definitive formulation, nor boundaries [16], thus the ability to analyze and solve them is very useful in real-life situations. In accordance with this view, Lu and Fletcher [37] proposed to teach CT by using languages based on notions that are familiar to people, rather than using programming languages; in this way, concepts like abstraction and algorithmic thinking could be more easily brought about.

[1] https://hourofcode.com

Repenning and colleagues modelled CT as an iterative process structured in three stages [47]: (1) *problem formulation (abstraction)*, namely a verbal or diagrammatic conceptualization of the problem; (2) *solution expression (automation)*, where the solution is described in a non-ambiguous way so that it can be executed by a computer; (3) *execution and evaluation (analysis)*, where one may obtain visualizations of the outcome from the other two stages and evaluate them. On the basis of the evaluation, problem formulation can be refined, and the cycle starts again. The idea proposed in [47] is that the three stages of the CT process should be supported and integrated by means of CT tools, such as any kind of programming (including EUD), but also informal drawings, mind maps, and task-specific languages.

2.2 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 [62]. In order to help to acquire CT skills two main approaches have been introduced in GBL: learning through designing games and learning through gameplay.

2.2.1 Learning through designing games

Learning through designing games has been studied for many years. In 1996 AgentSheets, designed by Alexander Repenning, was released, even though the first prototype dates back to 1989 [45]. This tool takes its name from the fact that the user develops the game on a grid resembling a spreadsheet, whose cells contain agents. These entities, visualized as pictures, can perform multiple actions like reading Web pages or playing sounds and animations. A graphical interface allows creating "if-then" rules that represent agents' behavior. In few training sessions, AgentSheets allows a middle-school student to design simple games, such as a Frogger-like game where a frog is controlled by the cursor, cars are moving and collisions between frogs and cars must be dealt with; but, with more training, it also allows creating highly sophisticated games with Sims-like behaviors [46]. It has been demonstrated that AgentSheets favors learning CT skills like problem-solving, abstraction, and pattern recognition [32]. Monteiro et al. [40] have recently studied how AgentSheets may improve algorithm design skills, thanks to the logic underlying the creation of rules, as well as teach the automation concept.

Alice is another visual programming environment, developed at the Carnegie Mellon University starting from 1997 [25]. It focuses on creating 3D programming projects consisting of adventure games, whose behavior is defined through a drag-and-drop interaction of specific blocks. By means of designing these games, users are exposed to basic programming concepts, both with imperative and object-oriented programming paradigms, without the burden of remembering syntactic constructs. Alice has been successfully employed to assess CT skills in primary and middle school [61, 63].

Similarly, Kodu is an Integrated Development Environment based on visual programming, which allows creating games structured as 3D worlds comprising different types of objects able to react to some events [21]. Like AgentSheets and Alice, Kodu is aimed at fostering familiarity with basic programming notions, in an intuitive and playful way. It supports learning notions related to the imperative programming paradigms, such as sequentiality, conditional instructions, variables and assignments; it also encompasses some concepts of object-oriented programming, such as that of classes, objects, and information hiding. Different studies have been performed to demonstrate the capability of Kodu

to improve CT skills such as problem-solving, abstraction, problem decomposition, and pattern recognition [20, 55].

Perhaps the most influential and versatile tool for learning how to program by designing games is Scratch,[2] developed at the MIT and publicly released for the first time in 2005. It is a visual programming language whose interaction is made simple thanks to draggable instructions represented by blocks, fitting one another like puzzle pieces. The process of assembling instructions is guided by the different shapes and colors of blocks, suggesting which constraints must be satisfied. One of its biggest strengths is the large and heterogeneous community of users that, combined with the possibility of reusing and remixing other users' code, permits to cooperate, share knowledge and realize complex games more easily. Scratch is widely considered a successful tool to teach programming to K-12 students and foster CT skills [5, 12, 23], even though some researchers underline that programming games with Scratch happens at a much lower code level than in other tools, such as AgentSheets [46].

Robot programming has also been regarded as a form of game-based learning. For instance, in [1] Lego Mindstorms[3] is used to improve CT skills of high school students. In this approach, the learner defines a set of steps (i.e., a program) that the robot must carry out to accomplish a goal; then, the robot follows the steps and executes the actions accordingly. The learner may debug the program by observing and tuning robot's behavior.

Tangible Programming is yet another way to create interactive games by mixing physical objects with a more traditional instruction-based approach. For example, in Tern [26] the goal of introducing children to computer programming is pursued by using small wooden cubes with instructions displayed on their faces. The sequence of instructions results in a series of movements performed by a small robot. In T-Maze [60] the programming phase is conducted in a very similar fashion, with a camera dedicated to reading the programming sequence in real-time. Children aged 5 to 9 years can create their own maze maps and complete escaping tasks, thus accomplishing simple programming tasks in an easy way.

2.2.2 Learning through Gameplay

Learning through gameplay (or Gameplay Learning) is another approach to GBL for improving CT skills. This approach encompasses a variety of serious games specifically developed to teach computer programming and foster CT skills. It might represent a valid alternative to learning through designing games, since, as highlighted in [33], building a game from scratch could be too challenging for novice programmers and thus frustrating for the majority of players.

Among them, [31] proposes Program Your Robot, a game prototype developed to support children in practising the five skills that the authors identified as fundamental for CT: problem-solving, algorithm design, debugging, simulation and socializing. It is a puzzle solving game in which the player has to assist a simulated robot to reach a certain point on a grid. Players thus design a solution algorithm that the robot will follow, by using symbolic representations of "action commands" (to move the robot in an environment) and "programming commands" (basic constructs such as sequence, selection, iteration, and function). These commands are dragged from their toolbars to specific areas of the environment. Players need to move the robot, activate lights and collect items by proceeding towards different

[2] https://scratch.mit.edu

[3] https://www.lego.com/en-us/mindstorms

levels of the game. Rewards are obtained in the form of new collectable items, slots or enemies to avoid as the player advances through the game. Program Your Robot is conceived as a serious game and thus differs from the software applications for game design mentioned previously, which can be deemed programming environments to all effects. Indeed, tools like Alice or Scratch were designed to teach the basics of programming and to show how fun it can be. Instead, Kazimoglu et al. [31] were moved by the goal of creating a game that could foster CT skills. However, also in Program Your Robot, the player is exposed to basic programming constructs and thus the gameplay keeps on being strictly related to a programming activity.

CTArcade [33] is another serious game where players have to design a set of rules that are executed by a character while playing Tic-Tac-Toe. Making this game rules explicit is considered an important process for improving CT skills; in fact, people often apply them in a natural, perhaps unconscious way and normally there is neither the chance nor the reason to transform this knowledge into abstract instructions. Lee et al. [33] report a study, implementing a "think aloud" protocol, where 18 children have been observed playing on CTArcade and on paper; the study shows that children articulate more CT skills (i.e. problem decomposition, pattern recognition, pattern generalization, and algorithmic thinking) using CTArcade compared to playing on paper. However, the analysis was carried out on Tic-Tac-Toe only, a widely popular and common game; therefore, CT was difficult to externalize and observe.

Liu et al. [35] investigate the use of TrainB&P to assist students in developing computational problem-solving abilities. With this simulation game, the students can construct a railway system and design the transportation behaviors of trains on a railway by using several building blocks such as straight, curved, and branch tracks. In particular, the system allows students to program the transportation behaviors and simulate them in a 3D environment. The results of the study carried out with the participation of 117 students, demonstrate how the gameplay enhances students' motivation to learn computational problem-solving and brings them in a flow state during the learning experience.

All the above systems use traditional interaction styles based on keyboard and mouse; on the contrary, even though TAPASPlay shares with them the objective of fostering CT skills through gameplay, it leverages an interaction style based on tangible objects and Virtual Reality (VR). Tangible interaction and VR have been chosen to increase the playfulness of the system and create an engaging and collaborative learning environment. Furthermore, our prior work on TUIs [57] proved that physical object manipulation might help users learn to deal with abstract concepts. Similarly, in a study with children aged 5-9, it has been demonstrated that tangible interaction has the potential to help children cultivate skills such as abstraction and problem decomposition [61]. In line with [30], TAPASPlay also aims to foster collaborative learning, that is, it regards CT as a creative and social practice. Finally, TAPASPlay fits within the realm of Constructionist Video Games [62], namely computational environments in which players may create personally meaningful artifacts to overcome artificial conflicts or obstacles resulting in quantifiable outcomes. In the following, the design and implementation of TAPASPlay are described in detail.

3 TAPASPlay

TAPASPlay is a turn-taking game intended for end-users with little or no experience in programming, who are trained in CT abilities to become able to participate in system design and EUD activities.

3.1 Design

TAPASPlay has been developed from TAngible Programmable Augmented Surface (TAPAS) [56], a TUI-based programming environment that allows end-users to build simple workflows by assembling different services with the aim of repurposing Pervasive Displays in the wild. The tangible interaction with TAPAS is carried out using smartphones as tangible objects and digital blocks projected over a tabletop surface; its interaction modality already provided interesting preliminary results [57] about developing CT skills, thus we decided to build on that to develop TAPAS even further.

As in TAPAS, interacting with TAPASPlay requires a Pervasive Display or a tabletop surface, an RGB camera and a smartphone. Smartphone movements on the display or surface are tracked by the RGB camera, which locates the position of a fiducial marker shown on the phone screen and uses it as a reference point. TAPASPlay has been implemented as a Web application that is projected on the tabletop surface and is able to interact with players' smartphones. A smartphone application provides players with additional feedback and tools for completing the game. Finally, VR technology is used to visualize the outcome of the game.

As previously mentioned, TAPASPlay can be regarded as a constructionist video game, where learners are engaged in creating artifacts personally meaningful to them [29]. The game aims at providing users with an educational and entertaining experience: the former has the goal of fostering CT skills, the latter is relevant to capture users' attention [41], thus encouraging users' involvement in the learning activity.

To foster CT skills, TAPASPlay, first of all, capitalizes on two main features of TAPAS, namely:

1. The interaction with the game is based on a puzzle metaphor that proved to be an intuitive approach to specify the solution to a given problem (i.e. algorithmic thinking) [57]. In particular, TAPASPlay communicates the existence of constraints and supports the gameplay through puzzle pieces and their shapes, aiding users whilst giving constraints in their selection process.
2. Puzzle pieces are physically manipulated, in order to favor the appropriation of abstract concepts through tangible interaction [61].

Furthermore, the gameplay is conceived around alchemy, that is the (fictitious) process of transmuting metals: players act as alchemists forging swords and shields, made of three different metals. In order to build each sword, players have a limited amount of energy points they can spend on transmutations, which in turn make a sword earn force points. Transmutations (also called transformations in the following) can be combined together in different ways, allowing users to experiment and practice problem abstraction and decomposition by satisfying the puzzle constraints. The objective is to maximise force points while carefully managing energy points on each sword. Trying to earn force points while finding a trade-off with energy points is a NP-hard problem that can be solved with different approaches (e.g., greedy algorithm, backtracking); this requires algorithmic thinking in finding even a sub-optimal solution.

To provide an entertaining experience, the game encompasses both competition and collaboration: the game runs in player versus player modality, but each character might be programmed by a group of players, thus favouring collaboration within the group. This allows us to accommodate different individual psychological needs. Indeed, according to Self-Determination Theory [49], humans are usually motivated to do something by the

needs of competence, relatedness and autonomy. A literature review about the use of games and game elements to facilitate learning [24] underlines how some works claim that competition can satisfy the player's need for competence [36], whilst a feeling of relatedness can be achieved for example in multiplayer games, that is, where players interact with one another [50]. In our case, relatedness is made stronger, because a group of players is called on to collaborate to achieve a common goal. Autonomy is finally achieved in TAPASPlay due to its constructionist nature since players are let relatively free to choose their sequences of actions to create the required artefacts (the swords).

Finally, the game features a storytelling suitable to a VR representation: the battle between the alchemists can be visualized by wearing affordable goggles (e.g., Google Cardboard[4]).

In summary, the game is structured in 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.

3.2 Forging swords

The first phase is aimed at fostering different CT skills, such as problem decomposition, algorithmic thinking, and abstraction. During the first phase, each player creates three offensive strategies by composing three different swords. In order to accomplish that, players have to attach transformations, represented as pieces of a puzzle, to a halo surrounding the user's smartphone on the main display. Each strategy is a sequence of transformations taken from a randomly generated set shown at the beginning of the game on the main display (Fig. 1).

Each half of the tabletop screen is available for a player to forge the swords. The halo, with its three hilts, follows the movement of the smartphone and, when a collision with a puzzle piece is detected, such piece is attached to the vertically oriented hilt given that the move is allowed by the game rules. The three swords are defined one at a time, so that each can have a different set of puzzle pieces available for the players, avoiding repetitions and increasing in difficulty. For instance, in Fig. 2, each player is creating his first sword.

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

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. 4a, it is equivalent to a transformation that turns bronze into steel. Each sword is made of a different type of metal, determined by the shape of the final puzzle piece. For example, in Fig. 4b the shape of the final piece is triangular and thus a steel sword has been forged.

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, each transformation consumes a number 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. 4a), and (4) the force points gained, displayed on the transformation as well (right half in Fig. 4a).

[4] https://vr.google.com/cardboard/

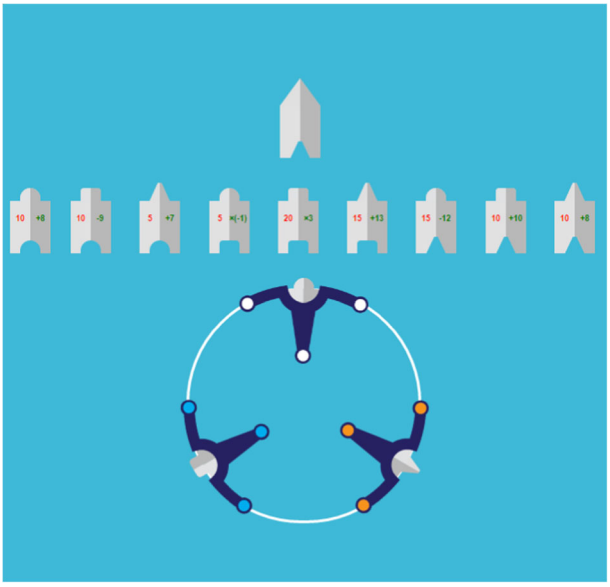


Fig. 1 Initial state of the game: the set of transformations is displayed, as well as the main halo with three hilts and the final piece

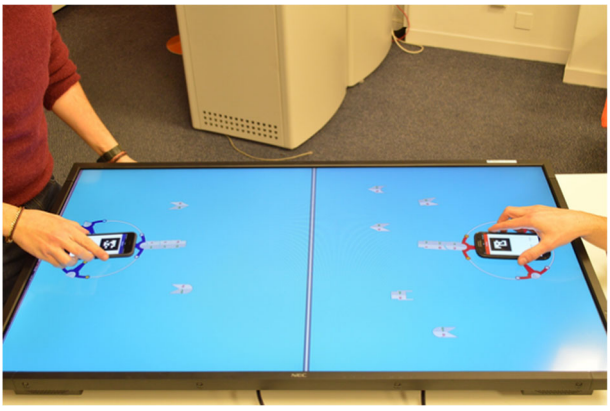
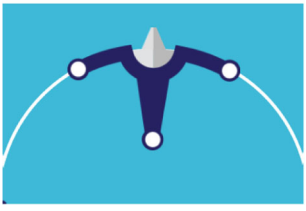


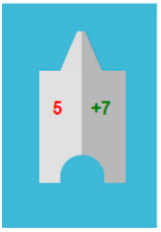
Fig. 2 Forging swords through tangible and puzzle-like interaction



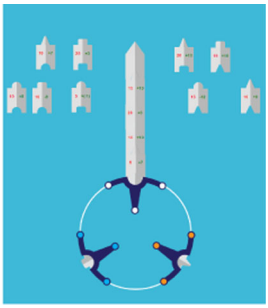
(a) An example of initial state of the sword.

(b) An example of final piece.

Fig. 3 Examples of initial and final pieces of a sword



(a) An example transformation.



(b) A sword example.

Fig. 4 Forging a sword by composing transformations

In order to apply a transformation, 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 sword (or, if the transformation applied is the first one, the input shape has to be the same as the output shape of the initial state of the sword); and (2) the alchemist must have an amount of energy points greater or equal than the one shown 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.

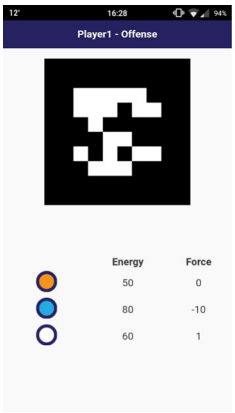
The initial state of each sword consists of an output shape attached to a hilt on the halo, a number of force points, and a number of energy points. The final state is reached when the player is satisfied with its sequence of transformations and decides to — and can — attach the final piece to the sword.

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 in the transformation. See for example Fig. 5a, 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.

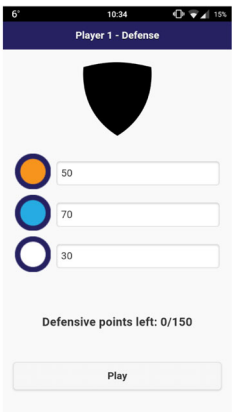
Maximizing force points requires to decompose the problem of forging a sword in smaller transformation problems (i.e. problem decomposition) and then solve sub-problems by selecting transformations through a greedy technique (i.e. selecting among the available pieces the one that gives more force points) or backtracking (i.e. going back to a previous decision point when reaching an invalid solution). During this activity, a player could mentally combine two transformations and regard them as a new piece with its own input and output shape, which can be used to forge the sword; therefore, abstraction comes into play during solution creation. Finally, the definition of each offensive strategy requires the player to iterate the steps of evaluation and selection of transformations until she is satisfied with the solution. The overall activity is then repeated three times, one for each sword, always with a game scenario (i.e. the available puzzle pieces) of increased difficulty.

3.3 Forging shields

The second phase is functional to the playability of the game and not strictly related to fostering CT skills, even though it requires some analytical abilities. In this phase, the



(a) The energy and force points of the swords.



(b) The defense points in the shields.

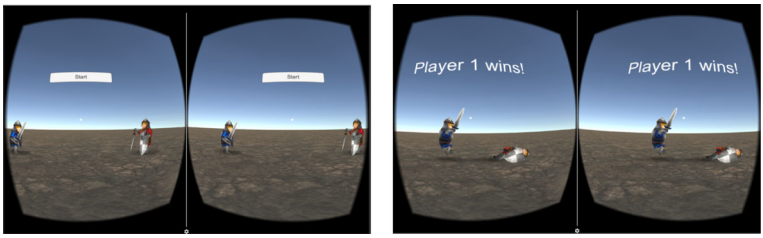
Fig. 5 The smartphone application

players must define their defensive strategies, which consist of allocating a number of defense points into three shields, each one corresponding to a different metal. The choice should be based on a couple of considerations: how the player guesses the opponent distributed force points on the different swords and which transformations have been chosen to build her own swords. For instance, if a player was not able to optimize the strategy for the steel sword, then she might consider allocating most defense points into the steel shield, in order to counterpoise her weak offensive strategy. To allocate defense points into the shields, each player interacts with a simple interface displayed on the smartphone (Fig. 5b).

3.4 Enjoying the battle in VR

The third phase of the game was designed to foster debugging capabilities, one of the main CT skills highlighted in the literature. In the current version of TAPASPlay, however, this feature is limited to the visualization of the battle in VR.

More precisely, when both the previous phases of the game are completed, a simple Android application showing a VR video is made available from the server. Both players must wear goggles (e.g., Google Cardboard) to enjoy the content of the video. The server provides each player with a different video on the basis of the scores it has received from the Web application. For instance, if Player 1, who used the halo with blue hilts, reached the highest score, the video shows a knight wearing a blue armour defeating the opponent dressed in red; otherwise, a video with reversed roles is played. The VR video shows two knights armed with sword and shield. In the beginning, a button with the "Start" label is visualized and a pointer placed at the center of the user's site suggests that gazing at it will allow playing the animation (Fig. 6a). After having pressed the button, the two knights approach the center of the scene and, when they are close enough, they start duelling. 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. 6b).



(a) Beginning of a duel. (b) End of a duel: player 1 has won.

Fig. 6 Visualizing the battle in VR

Supporting this VR visualization to inspect players' solutions provides an engaging way to visualise the outcome of the strategies and supports learning of debugging capabilities within gameplay.

3.5 Summary

Table 1 summarizes the aforementioned CT skills that the design of TAPASPlay strives to foster and the corresponding system features.

4 Exploratory study

This section presents the goals, hypotheses, and description of the user study we carried out, following the guidelines of the American Psychological Association [66].

4.1 Goals

The goal of our study is to evaluate whether TAPASPlay provides a fun and enjoyable experience to players while helping them develop CT skills through collaborative Gameplay Learning. The purpose is to evaluate how participants respond to playing with it and manage to build different strategies in order to win the game.

4.2 Research questions

User participation in system development can be effectively achieved by creating the conditions for their empowerment through a meta-design approach [17, 18] that supports them in appropriating those CT skills [64] necessary for understanding and contributing to systems evolution.

Gameplay offers an opportunity to promote high-level CT concepts indirectly in a playful way to an ever wider audience.

The main research question derived from this context is: *"Is gameplay an effective way to foster learning of CT skills?"*.

In order to effectively foster CT skills, gameplay should provide playfulness and collaborative learning. TUIs exploit our innate dexterity for objects' manipulation to aid understanding of abstract concepts — such as the ones involved by coding and CT [57]. Coupling them with gameplay also improves the playfulness [44], relieving users from the mental burden carried by more artificial interaction paradigms.

Table 1 CT skills considered in the design of TAPASPlay and related system's features

CT Skill	Description	Feature(s) in TAPASPlay
Problem-solving	Understanding what the goal of the problem is and finding a solution to achieve it	Defining a strategy that maximizes force points of each sword, given the energy points
Algorithmic thinking	A way of describing a solution to a problem through a set of steps	Forging a sword by combining puzzle pieces that represent metal transformations
Using abstraction	Identifying different levels of details of the problem	Mentally combining two or more transformations and regard them as a new one with its input and output
Using decomposition	Decomposing a complex problem into easier sub-problems	Decomposing the problem of forging a sword in the sub-problems of selecting appropriate transformations
Ability to think recursively	Decomposing a problem into smaller problems with the same structure as the original and making up the overall solution by combining the results of the smaller problems	Decomposing the problem of forging a word in the sub-problems of forging portions of the sword
Using heuristics	Discovering a solution through search and experimentation	Selecting the piece that gives more force points (greedy technique); going back to a previous decision point and choosing a different strategy (backtracking)
Data representation	Using different ways to represent different types of data and their relationships	Representing different types of puzzle pieces by means of their input and output shapes
Evaluating solutions (debugging)	Finding and resolving defects in a solution	Visualizing the battle in VR

The first sub-question is then *"Can we provide a playful way of learning CT skills?"*.

One of the other benefits of TUIs is their natural predisposition towards collaboration, which is beneficial not only for learning in general but especially for fostering CT skills [61]. Moreover, it is also a distinctive trait of some of the most engaging games.

The second sub-question is then: *"Can collaborative learning help improving CT skills?"*.

Playfulness demonstrated an effective strategy to learn programming skills (see Section 2.2.2) and our gaming sessions are aimed at testing whether it can be as effective for improving CT. Furthermore, we hypothesize that collaboration can play an important role in stimulating learning CT skills and in particular through the interactive tabletop collaboration. We hope that by studying how our platform provides a collaborative and playful experience for learning CT skills we contribute to widening end-users' participation in system design and development.

4.3 Study design

An exploratory research design was used, comprising think-aloud feedback, participants interactions recording and observation, and a post-test survey. Exploratory research is

Fig. 7 The study setting inside a university laboratory

usually conducted to determine the nature of a problem, thus it is not intended to provide conclusive evidence, but to achieve a better understanding of the problem [54]. We decided to conduct an exploratory study since fostering CT skills through gameplay has not been extensively studied in literature so far and we were interested in identifying issues and opportunities about how to design game-based learning systems that support learning of CT skills through gameplay instead of game design.

4.4 Participants

The participants of the study were 18 UK secondary school female students of Key Stage 4 (15 years old) coming from different schools of the London area. None of them had a solid programming background, but a small subset (3 of them) had a little experience in block-based programming with Scratch. No prerequisite knowledge was required to perform the tasks, and none of the participants had prior knowledge of either TAPAS or TAPASPlay. A brief introduction to the system and the game was provided to the group.

4.5 Settings and tasks

The study was conducted within the Brunel University London facilities, in a laboratory inside the Department of Computer Science, as depicted in Fig. 7.

We presented participants with TAPASPlay and tasked them with playing a single-turn game, i.e. forging one sword each; we purposively removed the VR visualisation and the defense strategy definition in order to focus our evaluation just on the proposed interaction and the effects of a TUI-based gameplay on CT skills.

The game scenario we developed (i.e. the available puzzle pieces at the beginning of a game, as reported in Table 2) was meant to provide players with many strategies to be discovered and different difficulty levels. This allowed us to investigate the level and progression of CT skills throughout the game, by assessing the difficulty of the issued strategies.

Indeed, the puzzle shapes provide constraints and introduce conflict within the game: a player needs to maximize the force points of her sword while using the available puzzle

Table 2 The TAPASPlay scenario we tested with participants

	Input shape	Output shape	Energy cost	Force points
1:	⋃	⊓	10	+8
2:	⋃	⋀	10	−9
3:	⋁	⋂	5	×(−1)
4:	⋁	⋀	20	×3
5:	⋁	⊓	15	+13
6:	⊔	⊓	5	+8
7:	⊔	⋂	15	−12
8:	⊔	⋀	10	+10
9:	⊔	⊓	10	+8

pieces in an appropriate order. Moreover, each puzzle piece has a cost (energy points), and the sum of the puzzle pieces' cost that make up a sword is bounded to 100. The initial and final shapes were both triangular.

4.6 Procedure

Participants were randomly clustered in 8 groups, 6 groups of 2 people each and 2 groups of 3, and the study was conducted in four different sessions, one for each individual game match played: one between the two teams of 3 people, and the rest between the other teams, paired in a randomised fashion.

All interactions with the tabletop surface and oral feedback provided during the game were recorded. At the end a random participant from each group was asked to fill in a short questionnaire about her experience. Responses were given in terms of (Q1) enjoyment, (Q2) collaboration, and (Q3) interactivity, using a Likert scale from 1 (strongly disagree) to 5 (strongly agree) — like similar well-known questionnaires (e.g., SUS [6]) — with a neutral midpoint (neither agree nor disagree) in order to avoid directing the choices towards just negative or positive sentiments. The questionnaire presented the following statements:

Q1 I enjoyed playing TAPASPlay.
Q2 I think TAPASPlay can be a fun game to play with friends.
Q3 I enjoyed playing TAPASPlay on a tabletop by moving a smartphone.

From an early internal playtesting phase of the game scenario we devised an average duration of each match of around 6 minutes. Each experimental session was then planned to last 15 minutes in total: 3 minutes for a brief explanation of how the game works and its rules, 2 minutes of practice, 8 minutes for the actual match, and 2 minutes to give feedback and fill in the questionnaire.

5 Results

We analysed the collected data by (1) performing a content analysis on the feedback and field-notes observations, (2) examining the recorded game strategies employed by participants, and (3) analysing the questionnaire results. We report our findings below.

5.1 Feedback and observations

The overall response was positive, but participants needed some practice at first to get going assembling swords. Indeed, all the groups managed to play and successfully assembly a sword in the given time.

All groups were involved in playing the game, and all groups members were trying to work out the right sequence of pieces to assemble the strongest sword possible. From the participants' observation, we also found that all participants offered their support and ideas to solving the problem and actively discussing within their group: none of the participants was left isolated. The collaboration aspect of TAPASPlay seemed to have appropriately worked to stimulate discussions and teamwork.

One interesting aspect concerned the interaction modality. The TUI seemed easily grasped and manoeuvred by participants, but the individual control point (i.e. the smartphone) and lack of support for mixed interaction (e.g., multi-touch) were pointed out by someone as the main hiccup to a better gameplay experience. A similar observation was also gathered in a previous study of TAPAS [39] employed in a different collaborative work scenario. Yet, in TAPASPlay the gameplay was devised to promote off-screen collaboration by allowing group members to interact with each other to reach a decision, and only one member to take control of the smartphone on the tabletop surface. In other words, this fostered group discussion and kept all team members involved in the decision process, balancing the need of each member to experiment with her own ideas and contribute to the overall discussion.

Another point that was made by some participants during the study was related to the fall-back mechanism of TAPASPlay. A simple undo action, triggered with a button on the smartphone interface, detaches the latest piece that was attached to the current halo and puts it back to its original position on the board, redistributing the energy points consumed. Three groups used the undo action, while the others preferred discussing the strategies amongst themselves and act once they figured out the whole process, without experimenting it first. Designing an improved fall-back mechanism that properly represents the system status and capabilities across different heterogeneous devices with a TUI is still an open question for Cross-Device interaction research [27].

5.2 Strategies

We recorded a total of 10 complete strategies — i.e. complete sequences of transformations from an initial to a target shape, as defined in Section 3.2 — produced by all the groups, as summarised in Table 3: on the left the strategies are reported as ordered sequences of the puzzle pieces in Table 2 with their corresponding numbers, as they were assembled during the game; the number of groups that issued a strategy is reported on the right when greater than 1. Six groups completed a single strategy each and decided to end the game there, while the other two groups — not playing in the same session — kept experimenting further after completing one sword, and issued two complete strategies each: the first successfully completed strategies (b) and (d), while the second (e) and (f).

The average number of pieces used to complete a sword was 3.4, with a standard deviation of 0.8. The majority of the strategies issued were naïve, in that they were built through a greedy algorithm using just a small number of pieces and without multiple trials (strategies (a), (b), and (c) in Table 3), while the other strategies were more complex and required more effort — i.e. backtracking — and discussion to be discovered.

Table 3 The strategies completed by participating groups

On the left, numbers correspond to the puzzle pieces labelled in Table 2, while on the right we reported the number of times (when greater than 1) the corresponding strategy was issued during the study

(a)	$5 \rightarrow 8$	
(b)	$5 \rightarrow 6 \rightarrow 8$	($\times 2$)
(c)	$5 \rightarrow 9 \rightarrow 8$	($\times 3$)
(d)	$5 \rightarrow 6 \rightarrow 9 \rightarrow 8$	($\times 2$)
(e)	$5 \rightarrow 9 \rightarrow 6 \rightarrow 8$	
(f)	$5 \rightarrow 9 \rightarrow 6 \rightarrow 8 \rightarrow 4$	

5.3 Survey

Lastly, the questionnaire was filled by a randomly chosen participant from each group, whose results are reported in Table 4.

All three statements proposed (Section 4.6) were rated positively by the majority of respondents. The game interactivity (Q3) was the most positively perceived with 7 out of 8 participants rating it positively, while 5 out of 8 participants judged the enjoyment (Q1) more than neutral. The collaboration aspect of TAPASPlay (Q2) seemed to have been appreciated by most of the participants, with 5 out of 8 positive, 2 neutral, and one negative rating.

6 Post-hoc analysis and lessons learnt

From the results collected during our exploratory study, we can discuss the Research Questions on designing a system to provide a playful way of learning CT skills and offering a collaborative platform that can help to improve CT skills.

We identified playfulness by analysing observations and feedback, together with the answers to the survey.

First, the experience provided by TAPASPlay was received positively from participants: the feedback recorded during the game reports a positive reception from users, which is also confirmed by the survey results (Q1). Devising an engaging gameplay is fundamental to foster CT skills, since it contributes to remove all the extra mental burden that comes from unnecessary game mechanics. TAPASPlay was also positively received in terms of interactivity, as evidenced by the survey results (Q3). TUIs provide a natural way of interacting with the game, without any artificial means of control, making the game easy to play and fun.

TAPASPlay also provides a new way of fostering CT skills through gameplay using a TUI that allows a highly interactive experience (survey's Q3) and at the same time well-crafted game mechanics. Indeed, all the groups completed at least one game with

Table 4 The survey results

	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
Q1	0	0	3	3	2
Q2	0	1	2	4	1
Q3	1	0	0	4	3

an appropriate strategy in the given time, demonstrating that through playfulness and collaboration CT skills can be rapidly introduced to the lay-person.

Collaboration is yet another aspect worth discussing in more detail, whose implications were identified through oral feedback, the questionnaire, and by analysing the observational data gathered during the study: TAPASPlay was designed from the ground up on top of TAPAS to support collaboration in order to foster CT skills. Indeed, Kazimoglu et al. [31] report that socialization is another CT skill fostered by learning through gameplay. The feedback obtained from participants and the results of the survey (Q2) confirms that TAPASPlay was well received as a collaborative game, stimulating discussions amongst team-mates and fostering an exciting learning environment. The gaming experience led users to socialize by continuously sharing thoughts about their approaches during the game, thus stimulating cooperative strategy development useful in co-design processes.

The lack of group members isolation is yet another benefit of the proposed gameplay observed during the study: softening the "lone wolf" effect — described as the preference to work alone and dislike of group processes — can positively affect team performance and improve learning [2]. Indeed, we seldom observe an even participation in learning groups, especially big ones [39], thus smoothing group participation level is a favourable consequence of balancing interaction style, groups activity, and size. Supporting mixed interaction modalities (e.g., multi-touch) and multiple control points within a single team as suggested by some participants might further improve the gameplay experience, but could also hamper the collaboration within a team by letting individual members explore strategies on her own. This aspect certainly deserves to be further analysed in the future.

Moreover, from the problem-solving strategies identified by participating groups we can also discuss how TAPASPlay fosters CT skills. Interestingly, the strategies adopted by the groups were quite different from each other, making use of a different amount of puzzle pieces; this demonstrates that TAPASPlay supports different types of algorithmic thinking (improving from a greedy approach to backtracking) on the basis of users' look-ahead and abstraction skills. Depending on the developed scenario, this can provide the right conditions for supporting CT skills at different levels, allowing players to identify an easy strategy soon and find the hardest ones as their skills progress (i.e. low floor-high ceiling [47]).

Another result worth pointing out is what happened to the two groups that issued more than one strategy (Section 5.2). The first one assembled strategy (e) in Table 3, then (f); this progression is evidence of a divide-et-impera approach, i.e. the result of decomposing a strategy into sub-problems and recursively solve them: once the problem has been solved with the first strategy, the group recognized that the solution could be extended by adding an extra piece, gaining more points. The second one assembled strategy (b) in Table 3, then (d); perhaps even more deeply than before, this progression is evidence of abstracting the building of a sword and recognizing that another piece can be added without changing the input and output of the whole strategy, thus completing it and gaining more points.

Indeed, by analysing those advanced strategies requiring more experimentation and discussion amongst players, there is an initial evidence that the collaborative aspect of our system and settings could provide an interesting effect on learning CT skills even in a limited amount of time.

All these 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. This way, we can ensure an active participation to system development and evolution of their users, aiding software designers in understanding and selecting the right solution while

helping to model the problem, thus coping with language and conceptual hurdles that often create a communication gap between them.

Such results were achieved within a group of young girls demonstrating that our design hypothesis on gameplay and collaboration can be positively exploited: gender imbalance and under-representation have always been major issues affecting the Silicon Valley and the whole tech community in general, making it necessary to come up with new strategies to correct this phenomenon [3, 11, 28, 58]. Engaging young girls in STEM activities means empowering them with the right tools to actively participate and take control of the issues coming up in the future, allowing them to take on a more central role in the science and technology sector. However, further studies are needed to validate the effects of TAPASPlay with a more diverse audience.

The study results also highlighted some of the current limitations of TAPASPlay that will need to be addressed in its future developments.

- The initial phase when the game rules were described and participants had a trial run to practice took around 5 minutes; considering that they were playing with a stripped-down version of TAPASPlay, we will need to consider ways of automating this task to free future evaluations from the presence of a researcher, which might limit their validity. An initial step-by-step tour of the game elements — perhaps controlled using the smartphone — might be helpful not only to describe rules and mechanics, but also in having users try out the tangible interaction before playing the game.
- As pointed out by some of the participants and noted from the study results, the fall-back mechanism of TAPASPlay needs some reworking: an improved undo action mechanism that properly represents the occurring changes of state in the system status and capabilities would help increasing the transparency and improving overall gameplay, even though it still represents a wider open issue for Cross-Device interaction research [27].

We can conclude by summarizing some early findings obtained in this exploratory study:

1. Playfulness plays a crucial role to involve a wide audience in learning CT skills. Embracing a playful experience can help students progress in STEM subjects by learning CT skills while enjoying the experience.
2. Designing a system to include collaborative aspects can be effective in stimulating end-user's reflections on problem-solving formalised through an interactive and collaborative technology, i.e. a tabletop. Those reflections can lead to learning quite articulate CT skills for problem-solving such as implicitly adopting a divide-et-impera strategy as unveiled during our study.

Finally, although preliminary and in an exploratory setting, we believe that our results can foster the design of a more detailed study looking at using measures of engagement and user experience [42] to inform the design of game-based tangible systems to foster a wider audience in learning CT skills.

7 Threats to validity

There are several validity threats to the design of this study.

Internal validity The limited number of participants allowed us to properly reason about different effects we found during the study, but a more extended experiment testing all the

game phases with more users needs to be designed in order to properly validate the survey results and the effects over isolation of team members, which cannot be definitive yet. The experimenter effect is concerned with any biasing effects in a study that is due to the actions of the researcher. The researcher attempted to carry out the study as much objectively and accurately as possible, acting as an observer limited to recording feedback and taking notes. The subject effect could have determined changes in the participants' behaviour due to being in the study and under observation; in our case, the study was carried out within a traditional learning environment during a series of workshops with similar game activities.

External validity Our sample was a group of only female students coming from different schools of the London area, which was a proper starting point to invastigate how to foster CT skills in an under-represented group within the tech industry, but in extending this work we should test TAPASPlay with a more diverse and international user group to investigate different effects. The lack of a mixed modality which fosters on-screen collaboration and support for an advanced fall-back mechanism limited in-game experimentation and prevented certain uses which might have affected the observed results.

Construct validity Due to experimenting time limitations, the post-test questionnaire was filled by a random member of each group and was limited to three questions designed to measure different aspects of the experience. This, together with the limited number of respondents, might have affected our results, even though we haven't used the survey results alone, but we cross-referenced them with the in-game oral feedback from participants.

8 Conclusion

The growing interest in CT is witnessed by very recent literature [68], which describes how CT is becoming more and more important in student and teacher education. In this paper, we claim that CT skills are fundamental to allow end-users to collaborate on system design and evolution at use time. For this reason, contrarily to other block-based approaches, in TAPASPlay blocks do not represent programming statements (as for instance the "if-then" block in Scratch) but remain at a higher level of abstraction, to promote problem decomposition abilities rather than programming ones.

Like TAPAS, TAPASPlay considers TUIs and physical object manipulation fundamental tools to make user activities more entertaining. Indeed, it has been demonstrated that tangible programming has the potential to help children cultivate skills such as abstraction and problem decomposition [61].

In this paper we presented the design rationale behind TAPASPlay, a turn-taking serious game using Gameplay Learning to foster CT skills by making learners' experience entertaining and social. In particular, we contributed to the research trend that explores learning through gameplay [31] — instead of learning through designing systems — in fostering CT skills. We tested our prototype with a group of secondary school girls and found evidence that TAPASPlay offers a playful environment to develop CT skills through collaborative learning.

In the future, we plan to focus on the concept of engagement and on how to improve the engagement in learning CT skills though TAPASPlay. To this aim, we will introduce additional features oriented to engagement and assess it by using appropriate instruments, such as the User Engagement Scale (UES) (in its long or short version) recently proposed in [42].

We are also planning to compare the results of this study against other games and digital tools designed to teach programming and foster CT skills to pinpoint those TAPASPlay features that could be used as a reference to design further games.

TAPASPlay is, however, a first proposal to fostering CT skills in end-users. Further experiments testing all three game phases with different user groups and game scenarios will be carried out in the next future to demonstrate the validity and robustness of the idea. Furthermore, several extensions of TAPASPlay have been already planned to tailor the system to end-users' characteristics and introduce different levels of complexity in the game. For instance, at the moment, only a VR simulation of the battle is available as an 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.

Acknowledgements The authors would like to thank Federico Danesi and David Bell for their contribution to the development of the system presented in this paper. We would also like to thank Stanislao Lauria and Lesley Warren for their help in coordinating the study. 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.

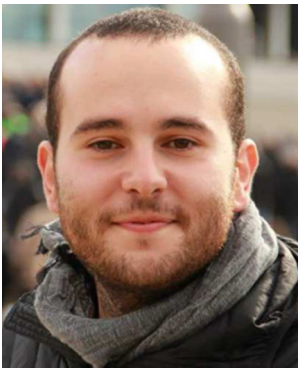
References

1. Atmatzidou S, Demetriadis S (2016) Advancing students' computational thinking skills through educational robotics. Robot Auton Syst 75(PB):661–670. https://doi.org/10.1016/j.robot.2015.10.008
2. Barr TF, Dixon AL, Gassenheimer JB (2005) Exploring the "lone wolf" phenomenon in student teams. J Mark Educ 27(1):81–90
3. Beckwith L, Burnett M, Grigoreanu V, Wiedenbeck S (2006) Gender hci: what about the software? Computer, 39(11)
4. Bers MU, Flannery L, Kazakoff ER, Sullivan A (2014) Computational thinking and tinkering: exploration of an early childhood robotics curriculum. Comput Educ 72:145–157. https://doi.org/10.1016/j.compedu.2013.10.020
5. Brennan K, Resnick M (2012) New frameworks for studying and assessing the development of computational thinking. In: AERA 2012
6. Brooke J (1996) Others: SUS-A quick and dirty usability scale. Usab Eval Indus 189(194):4–7
7. Bundy AL (2007) Computational thinking is pervasive. https://www.semanticscholar.org/paper/Computational-Thinking-Is-Pervasive-Bundy/6b7f70d987edc960d9864564dd4f6d121b647239
8. Burnett M (2009) What is end-user software engineering and why does it matter? In: Virtual and mixed reality. Springer, Berlin, pp 15–28, https://doi.org/10.1007/978-3-642-00427-8_2. http://link.springer.com/10.1007/978-3-642-00427-8_2
9. Cabitza F, Fogli D, Piccinno A (2014) Fostering participation and co-evolution in sentient multimedia systems. J Vis Lang Comput 25(6):684–694. https://doi.org/10.1016/j.jvlc.2014.10.014. http://linkinghub.elsevier.com/retrieve/pii/S1045926X14001074
10. Cabitza F, Fogli D, Lanzilotti R, Piccinno A (2016) Rule-based tools for the configuration of ambient intelligence systems: a comparative user study. Multimed Tools Appl 76(4):5221–5241. https://doi.org/10.1007/s11042-016-3511-2. http://link.springer.com/10.1007/s11042-016-3511-2
11. Cassell J, Jenkins H (eds) (1998) From Barbie to Mortal Kombat: gender and computer games. MIT Press, Cambridge

12. Cetin I (2016) Preservice teachers' introduction to computing: exploring utilization of scratch. J Educ Comput Res 54(7):997–1021. https://doi.org/10.1177/0735633116642774
13. Coutaz J, Crowley JL (2016) A first-person experience with end-user development for smart homes. Pervas Comput IEEE 15(2):26–39. https://doi.org/10.1109/MPRV.2016.24. http://ieeexplore.ieee.org/document/7445783/
14. Danado J, Paternò F (2012) Puzzle: a visual-based environment for end user development in touch-based mobile phones. In: Virtual and mixed reality. Springer, Berlin, pp 199–216. https://doi.org/10.1007/978-3-642-34347-6_12. http://link.springer.com/10.1007/978-3-642-34347-6_12
15. Dorner C, Hess J, ECIS VP (2007) 2007: improving information systems by end user development: a case study. aisel.aisnet.org
16. Fischer G (2017) Exploring design trade-offs for quality of life in human-centered design. Interactions 25(1):26–33. https://doi.org/10.1145/3170706
17. Fischer G, Giaccardi E (2006) Meta-design: a framework for the future of end-user development. In: End user development. Springer Netherlands, Dordrecht, pp 427–457. https://doi.org/10.1007/1-4020-5386-X_19. http://link.springer.com/10.1007/1-4020-5386-X_19
18. Fischer G, Fogli D, Piccinno A (2017) Revisiting and broadening the meta-design framework for end-user development. In: New perspectives in end-user development. Springer International Publishing, Cham, pp 61–97, https://doi.org/10.1007/978-3-319-60291-2_4. http://link.springer.com/10.1007/978-3-319-60291-2_4
19. Fogli D, Danesi F, Malizia A, Turchi T, Bell D (2017) Sustaining cultures of participation by fostering computational thinking skills through game-play. GHITALY@CHItaly. https://dblp.org/rec/conf/chitaly/FogliDMTB17
20. Fowler A, Cusack B (2011) Kodu game lab: improving the motivation for learning programming concepts. In: Proceedings of the 6th international conference on foundations of digital games, FDG '11. ACM, New York, pp 238–240, https://doi.org/10.1145/2159365.2159398
21. Fowler A, Fristce T, MacLauren M (2012) Kodu game lab: a programming environment. Comput Games J 1(1):17–28. https://doi.org/10.1007/BF03392325
22. Francese R, Risi M, Tortora G, Tucci M (2016) Visual mobile computing for mobile end-users. IEEE Trans Mob Comput 15(4):1033–1046. https://doi.org/10.1109/TMC.2015.2422295
23. Grover S, Pea R, Cooper S (2015) Designing for deeper learning in a blended computer science course for middle school students. Comput Sci Educ 25(2):199–237. https://doi.org/10.1080/08993408.2015.1033142
24. Grund CK (2015) How games and game elements facilitate learning and motivation: a literature review. In: Cunningham DW, Hofstedt P, Meer K, Schmitt I (eds) INFORMATIK 2015. Gesellschaft für Informatik e.V., Bonn, pp 1279–1293
25. Herbert CW (2010) An introduction to programming using Alice 2.2, 2nd edn. Course Technology Press, Boston
26. Horn MS, Jacob RJK (2007) Tangible programming in the classroom with tern. In: CHI '07 extended abstracts on human factors in computing systems, CHI EA '07. ACM, New York, pp 1965–1970, https://doi.org/10.1145/1240866.1240933
27. Houben S, Marquardt N, Vermeulen J, Klokmose C, Schöning J, Reiterer H, Holz C (2017) Opportunities and challenges for cross-device interactions in the wild. Interactions 24(5):58–63. https://doi.org/10.1145/3121348
28. Huff C (2002) Gender, software design, and occupational equity. SIGCSE Bull 34(2):112–115. https://doi.org/10.1145/543812.543842
29. Kafai Y, Resnick M (eds) (1996) Constructionism in practice: designing, thinking, and learning in a digital world. Lawrence Erlbaum Associates, Mahwah
30. Kafai YB (2016) From computational thinking to computational participation in k–12 education. Commun ACM 59(8):26–27. https://doi.org/10.1145/2955114
31. Kazimoglu C, Kiernan M, Bacon L, MacKinnon L (2012) Learning programming at the computational thinking level via digital game-play. Procedia Comput Sci 9:522–531. https://doi.org/10.1016/j.procs.2012.04.056. http://linkinghub.elsevier.com/retrieve/pii/S1877050912001779
32. Koh KH, Basawapatna A, Bennett V, Repenning A (2010) Towards the automatic recognition of computational thinking for adaptive visual language learning. In: Proceedings of the 2010 IEEE symposium on visual languages and human-centric computing, VLHCC '10. IEEE Computer Society, Washington, pp 59–66, https://doi.org/10.1109/VLHCC.2010.17
33. Lee TY, Mauriello ML, Ahn J, Bederson BB (2014) CTArcade: computational thinking with games in school age children. Int J Child-Comput Interact 2(1):26–33. https://doi.org/10.1016/j.ijcci.2014.06.003. http://linkinghub.elsevier.com/retrieve/pii/S2212868914000208

34. Lieberman H, Paternò F, Wulf V (2006) End user development (human-computer interaction series). Springer, Berlin
35. Liu CC, Cheng YB, Huang CW (2011) The effect of simulation games on the learning of computational problem solving. Comput Educ 57(3):1907–1918. https://doi.org/10.1016/j.compedu.2011.04.002
36. Liu D, Li X, Santhanam R (2013) Digital games and beyond: what happens when players compete? MIS Quart Manag Inf Syst 37(1):111–124. https://doi.org/10.25300/MISQ/2013/37.1.05
37. Lu JJ, Fletcher GH (2009) Thinking about computational thinking. SIGCSE Bull 41(1):260–264. https://doi.org/10.1145/1539024.1508959
38. Lye SY, Koh JHL (2014) Review on teaching and learning of computational thinking through programming. Comput Hum Behav 41(C):51–61. https://doi.org/10.1016/j.chb.2014.09.012
39. Malizia A, Turchi T (2015) Pervasive displays in the wild: employing end user programming in adaption and re-purposing. In: Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)
40. Monteiro IT, deCastro Salgado LC, Mota MP, Sampaio AL, de Souza CS (2017) Signifying software engineering to computational thinking learners with agentsheets and polifacets. J Vis Lang Comput 40:91–112. https://doi.org/10.1016/j.jvlc.2017.01.005. http://www.sciencedirect.com/science/article/pii/S1045926X16300234. Semiotics, Human-Computer Interaction and End-User Development
41. Moreno J (2012) Digital competition game to improve programming skills. J Educ Technol Soc 15(3):288–297. http://proxy.unibs.it/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=asx&AN=79816983&lang=it&site=eds-live&scope=site
42. O'Brien HL, Cairns P, Hall M (2018) A practical approach to measuring user engagement with the refined user engagement scale (ues) and new ues short form. Int J Human-Comput Stud 112:28–39
43. Papert S (1980) Mindstorms: children, computers, and powerful ideas. Basic Books, Inc., New York
44. Price S, Rogers Y, Scaife M, Stanton D, Neale H (2003) Using 'tangibles' to promote novel forms of playful learning. Interact Comput 15(2):169–185. https://doi.org/10.1016/S0953-5438(03)00006-7. http://www.sciencedirect.com/science/article/pii/S0953543803000067. Interaction Design and Children
45. Repenning A (2000) AgentSheets®: an interactive simulation environment with end-user programmable agents. Interactions. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.36.2039&rep=rep1&type=pdf
46. Repenning A, Webb D, Ioannidou A (2010) Scalable game design and the development of a checklist for getting computational thinking into public schools. In: Proceedings of the 41st ACM technical symposium on computer science education, SIGCSE '10. ACM, New York, pp 265–269, https://doi.org/10.1145/1734263.1734357
47. Repenning A, Basawapatna A, Escherle N (2016) Computational thinking tools. In: 2016 IEEE Symposium on visual languages and human-centric computing (VL/HCC). IEEE, pp 218–222. https://doi.org/10.1109/VLHCC.2016.7739688
48. Resnick M, Maloney J, Monroy-Hernández A, Rusk N, Eastmond E, Brennan K, Millner A, Rosenbaum E, Silver J, Silverman B, Kafai Y (2009) Scratch: programming for all. Commun ACM 52(11):60–67. https://doi.org/10.1145/1592761.1592779. http://portal.acm.org/citation.cfm?doid=1592761.1592779
49. Ryan RM, Deci EL (2000) Intrinsic and extrinsic motivations: classic definitions and new directions. Contemp Educ Psychol 25(1):54–67. https://doi.org/10.1006/ceps.1999.1020. http://www.sciencedirect.com/science/article/pii/S0361476X99910202
50. Ryan RM, Rigby CS, Przybylski A (2006) The motivational pull of video games: a self-determination theory approach. Motiv Emot 30(4):344–360. https://doi.org/10.1007/s11031-006-9051-8
51. Scaffidi C (2017) Potential financial payoffs to end-user developers. In: Barbosa S., Markopoulos P, Paternò F, Stumpf S, Valtolina S (eds) End-user development. Springer International Publishing, Cham, pp 183–190
52. Shneiderman B, Plaisant C, Cohen M, Jacobs S, Elmqvist N, Diakopoulos N (2016) Grand challenges for HCI researchers, vol 23, pp 24–25, https://doi.org/10.1145/2977645. http://dl.acm.org/citation.cfm?doid=2991131.2977645
53. Shute VJ, Sun C, Asbell-Clarke J (2017) Demystifying computational thinking. Educ Res Rev 22:142–158. https://doi.org/10.1016/j.edurev.2017.09.003. http://www.sciencedirect.com/science/article/pii/S1747938X17300350
54. Singh K (2007) Quantitative social research methods. Sage
55. Touretzky DS, Marghitu D, Ludi S, Bernstein D, Ni L (2013) Accelerating k-12 computational thinking using scaffolding, staging, and abstraction. In: Proceeding of the 44th ACM technical symposium on computer science education, SIGCSE '13. ACM, New York, pp 609–614, https://doi.org/10.1145/2445196.2445374
56. Turchi T, Malizia A (2016) Fostering computational thinking skills with a tangible blocks programming environment. In: 2016 IEEE symposium on visual languages and human-centric computing (VL/HCC).

IEEE, pp 232–233. https://doi.org/10.1109/VLHCC.2016.7739692, http://ieeexplore.ieee.org/document/7739692/
57. Turchi T, Malizia A, Dix A (2017) TAPAS: a tangible End-User Development tool supporting the repurposing of Pervasive Displays. J Vis Lang Comput 39:66–77. https://doi.org/10.1016/j.jvlc.2016.11.002, http://linkinghub.elsevier.com/retrieve/pii/S1045926X16302191
58. Tzafilkou K, Protogeros N, Karagiannidis C, Koumpis A (2017) Gender-based behavioral analysis for end-user development and the 'rules'attributes. Educ Inf Technol 22(4):1853–1894
59. Voogt J, Fisser P, Good J, Mishra P, Yadav A (2015) Computational thinking in compulsory education: towards an agenda for research and practice. Educ Inf Technol 20(4):715–728. https://doi.org/10.1007/s10639-015-9412-6
60. Wang D, Zhang C, Wang H (2011) T-maze: a tangible programming tool for children. In: Proceedings of the 10th international conference on interaction design and children, IDC '11. ACM, New York, pp 127–135, https://doi.org/10.1145/1999030.1999045
61. Wang D, Wang T, Liu Z (2014) A tangible programming tool for children to cultivate computational thinking. Sci World J 2014(3):1–10. https://doi.org/10.1155/2014/428080
62. Weintrop D, Holbert NR, Horn M, Wilensky U (2016) Computational thinking in constructionist video games. IJGBL 6(1):1–17. https://doi.org/10.4018/IJGBL.2016010101. http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/IJGBL.2016010101
63. Werner L, Denner J, Campe S, Kawamoto DC (2012) The fairy performance assessment: measuring computational thinking in middle school. In: Proceedings of the 43rd ACM technical symposium on computer science education, SIGCSE '12. ACM, New York, pp 215–220. https://doi.org/10.1145/2157136.2157200
64. Wing JM (2006) Computational thinking. Commun ACM 49(3):33–35. https://doi.org/10.1145/1118178.1118215. http://portal.acm.org/citation.cfm?doid=1118178.1118215
65. Wing JM (2010) Computational thinking: what and why? https://www.cs.cmu.edu/CompThink/resources/TheLinkWing.pdf
66. Wohlin C, Runeson P, Höst M., Ohlsson MC, Regnell B, Wesslén A (2000) Experimentation in software engineering: an introduction. Kluwer Academic Publishers, Norwell
67. Yadav A, Mayfield C, Zhou N, Hambrusch S, Korb JT (2014) Computational thinking in elementary and secondary teacher education. Trans Comput Educ 14(1):5,1–5,16. https://doi.org/10.1145/2576872
68. Yadav A, Gretter S, Good J, McLean T (2017) Computational thinking in teacher education. In: Emerging research, practice, and policy on computational thinking. Springer International Publishing, Cham, pp 205–220, https://doi.org/10.1007/978-3-319-52691-1_13. http://link.springer.com/10.1007/978-3-319-52691-1_13



Tommaso Turchi is a PhD Candidate at Brunel University London and a Research Associate at the same university since 2016. He received the Master's Degree with distinction in Computer Science from the University of Florence. His research interest includes Human-Computer Interaction, Artificial Intelligence and Information Retrieval. He is member of ACM (Association of Computing Machinery).

Daniela Fogli is currently associate professor at the Department of Information Engineering of the University of Brescia, Italy. She received the Laurea degree in Computer Science from the University of Bologna, Italy, in 1994 and the PhD degree in Information Engineering from the University of Brescia, in 1998. From 1998 to 2000, she was a post-doc grant holder at the Joint Research Centre of the European Commission. Since 2000 to April 2015, she has been assistant professor at the University of Brescia. Her research interests include meta-design, end-user development, web usability and accessibility, decision support systems. She is member of ACM (Association of Computing Machinery) and SIGCHI Italy (the Italian Chapter of ACM's Special Interest Group on Computer-Human Interaction).

Alessio Malizia is currently Professor of User Experience Design at the School of Creative Arts of the University of Hertfordshire and Adjunct Professor at the Faculty of Logistics of Molde University College. He moved to Hertfordshire from Brunel University London, where he was Senior Lecturer in Human-Computer Interaction. He has previously worked at Universidad Carlos III de Madrid, where he was Associate Professor of Human-Computer Interaction (HCI) and Social Computing, Sapienza University of Rome, IBM, SGI and Xerox PARC. His research and teaching interests focus on Human-Centred Systems. Alessio is interested in the design of Ubiquitous Interactive Systems with a special focus on the End-User Development community. He is particularly interested in systems where the physical and digital become seamlessly intertwined producing a new hybrid landscape and the study of problems arising from designing such complex hybrid environments involving collaboration of various disciplines and stakeholders.