Supporting Teachers, Engaging Students: A Collaborative Model for K-12 Computing Education

Entertainment Computing 54 (2025) 100937

Contents lists available at ScienceDirect

Entertainment Computing

journal homepage: www.elsevier.com/locate/entcom

Supporting teachers, engaging students: A collaborative model for K-12 computing education

Alberto Monge Roffarello *, Juan Pablo Sáenz *

Department of Control and Computer Engineering, Politecnico di Torino, 10129 Torino, Italy

ARTICLE INFO

Keywords:
4th-grade computing education
Project-based learning
Group work
Cross-age tutoring
Observational study
Co-design

ABSTRACT

Although the importance of Computational Thinking (CT) for children is increasingly recognized, its adoption in computing education curricula in primary schools is limited by several open challenges, including teachers' training and curricula development. Seeking to systematize a process that enables primary school teachers to teach CT through computing education in primary schools, we present the design, evaluation, and analysis of an introductory coding course for 4th-grade classes in a large Italian city, utilizing the Scratch platform. The course followed a project-based learning approach, empowering groups of children in designing and implementing simple video games, and explored the adoption of a collaborative strategy through which computing experts, class teachers, and high-school tutors proactively supported the project work. We evaluated and refined the course educational strategies by conducting an observational study and co-designing activities with the involved teachers. Then, we derived an educational model that may allow K-12 teachers and experts to collaborate in designing and implementing computing education courses that are engaging, inclusive, and supportive.

1. Introduction

Over the past few decades, many countries have recognized the growing importance of technology and digital literacy, highlighting the need to foster Computational Thinking (CT) among students [1–3]. Using CT language in K-12 pedagogy, in particular, enables students to understand computational processes and develop skills for abstracting and representing information [4]. Yet, successfully integrating CT into school computing education curricula poses two well-acknowledged challenges [5]. First, teachers often lack proficiency for integrating and conducting CT activities into their classrooms [5,6] (*teacher training* issue). Second, such a lack of proficiency means that computing education curricula in K-12 education are often bounded to traditional approaches only, *e.g.*, unplugged coding, and effectively motivating and engaging children becomes challenging (*curricula development* issue).

Although there are individual research efforts and documented experiences that aim at overcoming the above mentioned issues, less attention is paid to abstract such learning lessons into comprehensive and actionable implications. This led us to formulate our research question: what strategies and considerations can guide primary school teachers to *proactively*, *engagingly*, and *supportively* integrate CT into their computing education curricula? To answer this question, we conducted a three-step work involving different actors (see Fig. 1 for a summary of the different phases and contributions). Specifically, we report on the design of an introductory coding course for 4th-grade classes (A), a two-months-long observational study (B), and a set of a co-design activities involving K-12 teachers (C), showing how this journey led us to define an educational model that may support the integration of CT into primary schools' computing education curricula in a proactive and supportive way (D).

The introductory coding course was designed by blending established educational strategies with innovative approaches to teach and support children, involving various actors proactively. Specifically, the course relies on a *project-based learning* approach through which *groups of children* get proficient in computational thinking through the design and implementation of simple video games in lessons that are *co-conducted* by experts and teachers. Furthermore, it involves external actors, *e.g.*, high-school students, taking the role of *cross-age tutors* who proactively support children during the experimental activities, and *training sessions* with teachers and tutors to enable them to collaborate in implementing the course.

To inform the development of an integrated model based on the course experience, we first conducted an observational study in two 4th-grade classes of a deprived neighborhood of a large Italian city. During the observational sessions, we analyzed whether children could fulfill each project step working in a group, measuring the deviations

* Corresponding authors.
E-mail addresses: alberto.monge@polito.it (A. Monge Roffarello), juan.saenz@polito.it (J.P. Sáenz).

https://doi.org/10.1016/j.entcom.2025.100937
Received 22 November 2024; Received in revised form 30 January 2025; Accepted 5 March 2025
Available online 14 March 2025

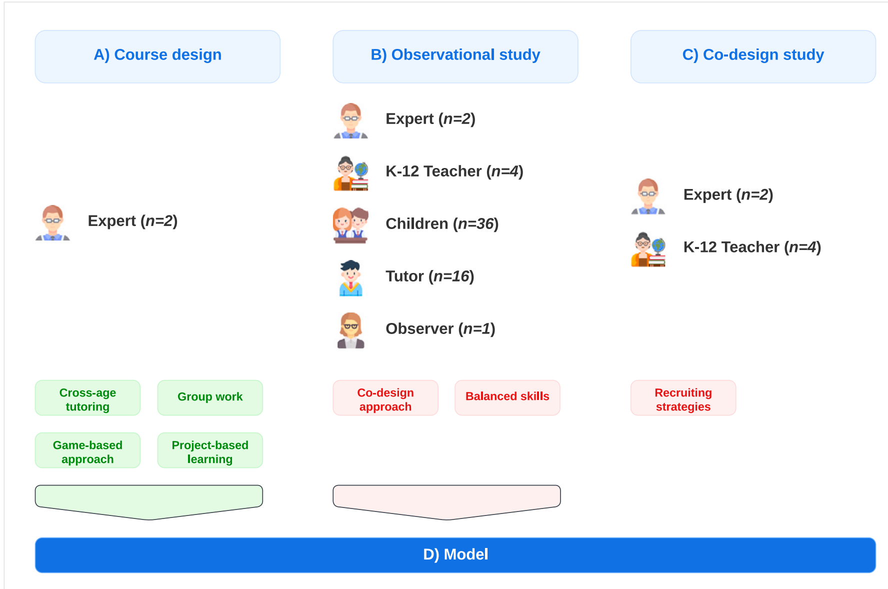


Fig. 1. A summary of phases, actors, and outcomes involved in our work.

from the examples shown at the beginning of the experimental activities. Furthermore, particular attention was devoted to observing how the different actors, *i.e.*, experts, teachers, high-school tutors, and children, interacted. We noted all the successful and unsuccessful cooperations between high-school students and children, as well as the degree of involvement of the class teachers in the experimental activities. Overall, the collected results highlighted the need for greater involvement of class teachers, the importance of creating groups of children with similar skills, and the importance of having highly motivated high-school students who take a proactive tutoring role.

Based on the observational study outcomes, we identified the need to involve teachers in designing the course rather than relying exclusively on training activities. As such, we tried to address this need by conducting a set of co-design sessions with four teachers participating in the first-course edition. During these co-design activities, the four teachers were asked to reflect on the needs, feelings, and challenges they experienced during the course, and, accordingly, to design and implement a new gamified project to be adopted in the course. The participating teachers successfully designed and implemented two video games that closely align with the school's curricula goals, highlighting the importance of having an educational model that fosters greater involvement from classroom teachers in shaping educational experiences.

We condensed our design decisions and findings from the observational study and the co-design experience into an educational model. The model specifically addresses the *curricula development* and *teacher training* issues, allowing K-12 teachers and experts to collaborate in the design and implementation of computing education courses that address CT engagingly, inclusively, and supportively.

2. Related work

2.1. Challenges in integrating CT in primary education

In today's digital age, computing has become an essential part of our daily lives, and its importance is continuously growing. As such, computing is increasingly important in K-12 education, enabling students to develop skills and knowledge essential for success in the digital age [3,7]. The central role of computing education in primary and middle schools is to provide students with the essential skill of *Computational Thinking* (CT) [8], *i.e.*, the transversal ability to understand and solve complex problems thinking like a computer scientist [9]. Computing education in K-12 can be implemented in various ways, such as through standalone computing courses, integrating computing concepts in existing curricula, and extracurricular activities such as coding clubs. In Italy – the country where our work has been conducted – computing education has recently gained importance thanks to the 2015 "Buona Scuola" reform [10], which introduced coding and robotics curricula in primary schools. Despite these efforts, Di Pietro et al. [6] identified obstacles hindering the adoption of computing education in Italian primary schools, particularly in rural and suburban contexts. In particular, although tenured Italian teachers can attend extra-curricula courses about computing education, such a topic is only marginally covered during their university studies [11]. As a result, teachers most often exclusively focus on unplugged coding, *i.e.*, the process of teaching computer programming concepts without the use of any digital devices [12], or slavishly follow existing initiatives and frameworks, *e.g.*, the "Hour of Code" initiative by *code.org*,[1] [13]. For example, Corradini et al. [13] examined the impact of the "Programma il Futuro" initiative, which aims to promote CT among Italian students. The study used quantitative and qualitative methods to analyze the Hour of Code's effectiveness after two years of implementation. The results indicate that the program has positively impacted students' CT skills. However, the authors also found that following a "closed" and fixed teaching path – the one proposed by the Hour of Code – may hinder teachers' and students' creativity. To overcome this issue, our work explores the adoption of a project-based collaborative approach to support children in creatively exploring computing-related concepts, enabling them to design and implement their own video games without being forced to pursue a unique, fixed solution. Although our work is based in Italy, it is worth noting that the challenges mentioned above are common worldwide and closely reflect the two main obstacles described by the "Informatics for All" initiative devised jointly by ACM Europe and Informatics Europe [5], *i.e.*, teacher training and curricula development.

The following sections review documented experiences from the computing education literature that address these challenges. However, translating this research into actionable implications that a wider audience can easily understand and apply remains a challenge. As such,

[1] https://code.org/ last visited on February 15, 2023

our work aims at deriving a model that empowers primary school teachers to *proactively*, *engagingly*, and *supportively* integrate CT into their curricula.

2.2. Teachers training and co-design

Our work reinforces the critical need for training teachers on how to integrate CT into K-12 computing education, as highlighted by a growing body of research.

Yadav et al. [14] found that introducing CT concepts into a required educational course for pre-service teachers positively impacted the teachers' understanding on how to incorporate CT concepts in K-12 classrooms. In particular, teachers who underwent CT training began to see it as a problem-solving approach that uses algorithmic thinking and abstraction, unlike those who hadn't received training, who thought CT just meant using computers. In the same vein, Simmonds et al. [15] designed a 12-hour workshop to train the K-8 teaching staff in a rural school. Their goal was to help these teachers develop basic CT skills and devise new ways to incorporate what they learned in their classrooms in the context of rural and vulnerable populations. The authors draw attention to the importance of addressing the barriers to entry by designing a national strategy for introducing CT in the classroom. This training program should boost the self-confidence of both teachers and students, offer an enjoyable experience, and align with the students' interests and requirements. This national strategy should also define a support structure involving researchers and universities to ensure replicability. Ketelhut et al. [16], instead, focused on integrating CT into elementary science instruction. They introduced a professional development program for mentor teachers to address the hesitation of pre-service teachers in adopting CT innovations. Despite acknowledging CT's benefits, mentor teachers faced challenges in effectively integrating it into the curriculum, advocating for CT in unfamiliar environments, not prioritizing science, and accessing the necessary resources and support for implementing CT in their teaching.

Similar to these works, in which various types of educators interact, our study involves computing experts and K-12 teachers. To further solve the teacher training issue, however, our work also empathizes the possibility to empower teachers in co-designing the CT teaching experiences. Using co-design can play a crucial role in ensuring that computing education courses meet the needs and interests of primary school students. Specifically, co-designing activities involving experts and teachers allow teachers to internalize better and learn computing concepts while ensuring that computing education courses for primary schools are adequately designed from a technical point of view. The key to co-design is that users and stakeholders are treated as partners in the design process rather than simply as sources of information [17,18]. Our approach was informed by the work of Kelter et al. [19], who discussed the initial iteration of the CT Summer Institute, a month-long program in which high school teachers collaborated with researchers to co-design math and science curricula incorporating computational elements. Co-design between teachers and researchers showed promise in enhancing teacher design capacity, aligning with broader curricular reform and professional development goals.

2.3. Engaging students

In addition to empowering teachers, our work also strives to effectively motivate and engage children in pursuing computing education courses through the adoption of group work and game-based approaches, providing students with a proactive and tailored support.

Papavlasopoulou et al. [3] described a two-year design-based research (DBR) project during which the authors conducted workshops where students, in groups, used Scratch and collaboratively created a socially meaningful artifact as a videogame. The authors determined that social engagement is essential as students work in front of the computer and reflect on their progress as a team, sharing the same goal to create an artifact successfully. In this regard, our proposed model is designed to incorporate group work among the children as a means to continually gather feedback, enhance their self-confidence, and stimulate discussions on strategies to address encountered challenges.

Regarding game-based approaches, Videnovik et al. [20] conducted a comprehensive literature review to examine the current trends and research gaps in game-based learning in computer science. The results show that the number of research articles has increased through the years, confirming the importance of implementing a game-based approach in computer science. Most of the analyzed articles implement a game-based approach through learning by playing, and no significant focus is given to the effectiveness of learning by designing a game as a pedagogical strategy. Furthermore, while the reviewed articles focus on using game-based methods to teach programming, there is a noticeable absence of studies detailing the development of students' knowledge and skills in various other computer science topics through game design. Consequently, the authors suggest that the lack of research articles applying a game-based approach to teaching diverse computer science subjects in primary education represents a promising avenue for future research. Our work is situated within this research domain, with the objective of encapsulating our experience into a model that can provide guidance for designing and implementing educational experiences rooted in a game-based approach.

Finally, in attempt to further support CT integration into schools' curriculum, we also involved high school students who were capable of providing proactive support to children. As we will describe later in this article, we consider that this cross-age tutoring strategy [21–23] has the potential to foster a relaxed and supportive environment, encouraging children to seek assistance without fear of judgment. As suggested by Tenhovirta et al. [23], cross-age peer tutoring practices offer substantial assistance to teachers, enabling them to concentrate on flexible pedagogic orchestration while also preventing them from being overwhelmed by the technological demands of the various digital tools used. Furthermore, tutoring can facilitate mutual interaction between the tutor and tutee, especially considering that an adult may be perceived as too distant in terms of authority and knowledge by the children [23].

3. Introductory coding course

We designed an introductory coding course exploiting the Scratch platform to foster CT while overcoming the challenges characterizing computing education in primary schools described by the "Informatics for All" initiative [5], *i.e.*, teacher training and curricula development. We selected Scratch as it has been identified as an innovative and effective tool to support CT in K-12 education [24,25]. At the same time, prior work suggests that Scratch helps teachers understand basic computing concepts significantly better than traditional programming languages [26], thus potentially empowering teachers to master and conduct computing education courses even without possessing a strong technological background.

We designed the course targeting 4th-grade primary schools, leveraging our own teaching experience – especially at the university level [27] – and previous works in K-12 education. The course employs a mix of established educational strategies and innovative approaches to teach and support children, involving various actors proactively. Specifically, the following are the three main course's ingredients and the rationale for which we chose them:

- The designed course follows a **project-based learning approach**, with children that incrementally implement a video game using the Scratch platform **working in group**. Group work – also known as cooperative learning [28] – is a strong incentive for learning at all levels in most educational systems [29], including in primary schools [30]. Paired with project-based learning [31] and gamification [32] – the development of a video game, in our

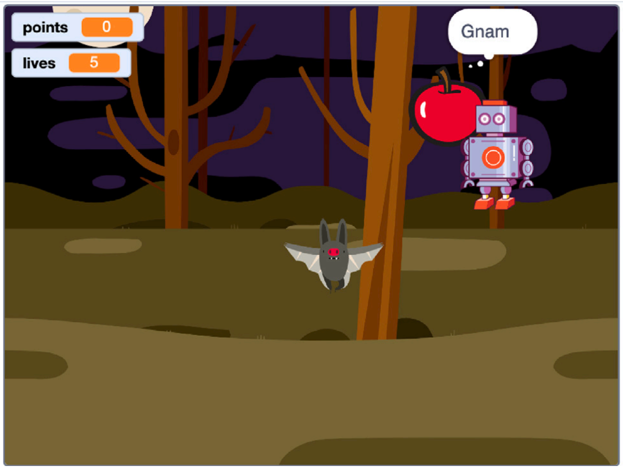


Fig. 2. A screenshot of the "Catch the Apple" game used as a reference in our introductory coding course.

case – it empowers students to work in groups trying to solve a specific problem, thus requiring critical thinking, collaboration, communication, and creativity. With such an approach, we envisioned a computing education course as interactive as possible, aiming to minimize the curricula development issue.

- The designed course follows is jointly conducted by **computing experts and class teachers**, who are instructed to favor a **trial-error dynamic** through which the 4th-grade students can constantly assess their progress, discuss and refine their implementation, and derive the final working solution. Experts' and teachers' responsibility, in particular, is to guide the children's analytical process, understand their reasoning, and foster or correct it but never reveal the solution. Trial and error [33] is a common approach used in computing education, as it can help students develop problem-solving skills and better understand the underlying concepts and principles. To the best of our knowledge, having this approach co-conducted by computing experts and class teachers is a novel aspect of our course. On the one hand, experts can help teachers when they encounter difficulties in managing and explaining computing concepts (teacher training issue). On the other hand, class teachers can ensure that the course and its progress over time are tailored to the skills and capabilities of their students (curricula development issue).
- The designed course adopts a **proactive cross-age tutoring approach** to proactively support children in developing their projects. In particular, each group of children is supported by a high-school student who interacts with children during the implementation of their game by giving them suggestions and directions and, most of all, posing questions to foster their analytical and computational thinking competencies. Although there are examples of cross-age tutoring in K-12 education [34], its application in our course is particularly important to offer each group of students a personalized experience according to their skills and capabilities, overcoming the limits of traditional educational approaches (curricula development issue).

Table 1 summarizes the six one-and-a-half-hour lessons that compose the designed course, along with the learning objectives addressed and the project steps that the children's groups must achieve. In particular, an example video game named "Catch the Apple" exemplifies and defines each lesson's expected outcome (Fig. 2). The game consists of a hero that must catch, one by one, ten apples that randomly appear on the screen while avoiding being bitten by a bat. However, this script is by no means a strict specification but just a high-level schema. In practice, children are encouraged to design and implement their own video game, *e.g.*, by choosing the main character (the hero), the reward (the apple), and the villain (the bat). As discussed in Section 4, this flexibility led to children inventing various plots, adding new characters, and customizing the graphical elements. In Lesson 1, children are introduced to the Scratch programming environment and can choose and customize their hero, villain, and reward (learning objective: *IDEs and coding environments*). Then, each lesson requires applying new, more complex computational concepts and code constructs. In Lesson 2, children learn how to *handle events*, associating the arrow keys to the hero's movement. Lesson 3 is instead focused on *loop statements*, with children who learn how to make their evil character move continuously on the screen. In Lesson 4, the learning objective is about *conditional statements*, as the required project step is to make the reward appear in a random position when the hero touches it. Lesson 5 is instead dedicated to *variables*, *i.e.*, points and lives, that change depending on the player outcomes. Finally, the last lesson is dedicated to presenting the outcomes of each group to the class.

Before implementing the course, class teachers and high-school students are requested to attend a 4-hour training session during which experts introduce them to their role in the course and the educational strategies to be followed. Furthermore, teachers and high-school students are asked to implement the "Catch the Apple" game on Scratch while experts explain the related programming concepts.

4. Observational study

We evaluated our introductory course by conducting an observational study to understand whether and how the adopted educational and supportive strategies – from the project-based learning approach to the proactive support by high-school students – worked. Observational studies are a well-established research method in HCI that entail observing and recording people's behavior in their natural settings with little or no interference from the researcher [35].

4.1. Methodology

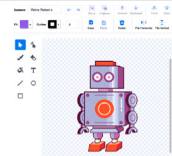
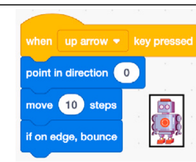
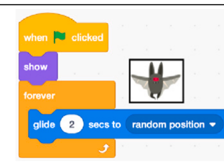
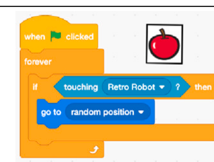
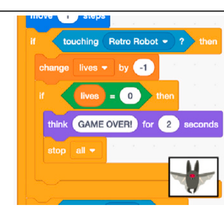
4.1.1. Observed classes and participants

The conducted observations were framed in the project *"Batti il 5",* [36] an Italian national project that aimed to prevent and counteract educational poverty by promoting the development and enhancement of activities in science, technology, and digital areas by introducing experimental environments, such as knowledge and curiosity labs, science and measurement labs, English for all, or coding and robotics labs. We conducted naturalistic observations of the devised computing education course in two 4th classes (CLASS A and CLASS B) of a primary school located in a deprived neighborhood of a large Italian city, with a total of 36 students observed (18 children for each class).

There were no cases of disabilities or specific learning disorders in CLASS A. CLASS B, instead, had a child with a selective mutism disorder and a child with Down syndrome. Selective mutism [37] is characterized by a persistent inability to speak in particular situations. Children with this disorder can understand spoken language and have the ability to speak normally, but the latter is manifested only in the home environment. Down syndrome [38], on the other hand, is a genetic abnormality condition that causes permanent intellectual disabilities and growth retardation, in addition to other health problems. Individuals with Down syndrome often exhibit distinctive physical features such as a wide face and forehead, weak muscle tone, and shorter-than-average legs and arms. This condition affects both children and adults.

Regarding the technological equipment, each group of children had a notebook with the Scratch program installed at their disposal. At the same time, experts and class teachers could use an interactive whiteboard.

Table 1
Lessons' description and involved programming concepts.

Lesson content	Learning objective	Project step
Lesson 1: Scratch is introduced, and children are shown an example of the video game they will implement, *i.e.*, the "Catch the Apple" game. Then, they learn how to create the Sprite, which represents their game's hero, reward, and villain, and how to set up a background.	IDEs and coding environments	
Lesson 2: Children learn how to make the hero move around the screen via the keyboard's directional keys.	Events handling	
Lesson 3: Children make their evil character appear and move continuously and randomly along the screen.	Loop statements	
Lesson 4: Children create rewards that appear on the screen in random locations and stay there until the hero touches them.	Conditional statements	
Lesson 5: Children work on the points and life management through variables: when the hero touches a reward, he earns one point. Conversely, when touched by the evil character, he loses a life. The game ends when the hero loses all ten lives or has collected all ten rewards.	Variable declaration and management	
Lesson 6: Each group comes to the front and presents their game to the class.	–	

Regarding the course settings, each lesson was jointly conducted by a class teacher and a computing expert (one of the authors of this paper). Overall, 2 class teachers (one for CLASS A and one for CLASS B) were involved in the course and participated in at least one lesson. These teachers had some basic knowledge of the Scratch platform, but only one had conducted some offline activities related to coding during their regular classes. The expert introduced each lesson and defined the expected outcomes. Furthermore, the expert could also intervene during the lessons to show examples on the whiteboard and provide students with hints on how to solve a given project step. Instead, the class teacher helped the expert explain expected outcomes and possible solutions to the children, *e.g.*, by using languages and metaphors appropriate for their students.

Student groups were composed of two children, each with the proactive support of a high-school student recruited from a technical high school located in the same primary school district. Two teacher's aides supported the two children with selective mutism disorder and Down syndrome in CLASS B.

4.1.2. Procedure and analysis

The observational study started in January 2020 and lasted two months. To avoid being intrusive or modifying behaviors and minimize the Hawthorne effect [39], we adopted a complete observer strategy through which observations were made by an external and detached observer (a graduate student in primary education sciences). During the observational sessions, the observer never interacted with any of the subjects in the class. The observer took notes on a personal computer for later analysis, using an observation sheet for each lesson. The sheet included a field to describe the aim of the lesson and another field for the observer to freely add any aspect emerging from the in-class activities that were considered relevant.

After the observational sessions, we held weekly meetings with the observer to analyze the collected data, evaluate the course strategies, and find possible improvements. Then, we – the two authors of this paper – merged all the observations collected by the observer in the Taguette application [40,41], and coded them using a deductive coding approach. Deductive coding is a widely used qualitative research method for determining the prevalence of pre-defined themes across data [42].

To ensure a systematic and reliable analysis, we followed a structured coding process. First, we familiarized ourselves with the observer's notes by repeatedly reading them to gain an overall understanding of emerging themes. Next, we applied the deductive coding approach by independently annotating excerpts according to the predefined codes. To improve reliability, both authors coded a subset of the data separately and then compared their results, discussing discrepancies until a consensus was reached. After establishing consistency, one author coded the remaining data, with periodic cross-checks by the second author. Following the initial coding, we conducted a thematic analysis to identify patterns and relationships within the data. We grouped excerpts under each code and iteratively refined subcategories that emerged within the three main themes (Student Progress, Interaction Between Actors, and Teachers' Involvement). These themes were directly linked to our research question on systematizing a process that empowers primary school teachers to *proactively*, *engagingly*, and *supportively* integrate CT into their curricula.

The codes, along with the rationale, were:

Student Progress: how students *progressed* in their projects, *e.g.*, by deviating from the examples shown at the beginning of the experimental activities. Our initial hypothesis was that high levels of deviation suggest students are proactively using CT skills to tackle challenges in their own way. This indicates a successful approach in engaging students.

Interaction Between Actors: any relevant aspects related to the *interaction* among the different actors involved in the course, *i.e.*, experts, teachers, high-school tutors, and children. Our initial hypothesis was that strong collaboration and knowledge exchange between teachers and other actors support a proactive and supportive environment. This can empower teachers to implement CT confidently in their own classrooms.

Teachers Involvement: the degree of *involvement* of the class teachers in the experimental activities. Our initial hypothesis was that a high level of teacher involvement beyond simply observing suggest their commitment to learning and implementing the CT curriculum effectively. This aligns with the research question's focus on proactive and supportive integration.

The following are two examples of how the observer's notes were coded using the designated coding scheme:

> *"Children are starting to explore Scratch, and I'm starting to understand the dynamics present in certain groups. In one of these groups, a child is having trouble using the 'point in direction' block. In fact, if the hero is not oriented in the intended direction, it continues to move in the wrong direction. In another group, children are interested in incorporating a challenging feature that is not part of the lesson"* (**code: student progress**).

> *"Two groups exhibit high levels of distraction, resulting in a chaotic and disorganized approach to problem-solving. The student tutor assigned to the group fails to assist them in maintaining focus on their tasks, but instead contributes to their distraction" (**code: interaction between actors**).*

Finally, we put together all the observations with the same code, trying to extract positive and negative implications.

4.2. Results

During the observed lessons, all the groups successfully created their own video game with the help of experts, teachers, and high-school students. After some explorations and hints by the expert and the proactive tutors, in particular, most of the students successfully understood the basic computing concepts they need to use, *e.g.*, loops and conditional statements. All the groups deviated from the examples shown at the beginning of the course by selecting their own characters, backgrounds, and objectives (see Fig. 3 for an example).

Overall, these were the main positive aspects that we coded on the collected observations:

- **(+)** Children appreciated the approach of creating a video game: they demonstrated interest in this kind of activity and creatively customized the showed examples to create their own solutions. Many groups, for example, carried out extra activities not explicitly requested by the experts and teachers and developed additional features in their video games with the help of high-school students (code: *students' progress*).
- **(+)** The explanations provided by the expert were clear and easy to understand. The expert's support was particularly helpful in supporting children groups in adopting a trial-and-error approach: the expert moved between the desks and provided students with hints on how to proceed or fix potential errors without disclosing the complete solution (code: *interaction* between experts and students).
- **(+)** As the class teachers know the specific skills of each student, their support was most often effective in motivating children to follow the expert instructions and recover from errors (codes: *interaction* between teachers and students, *teachers involvement*).
- **(+)** To a large extent, high-school students created a fruitful relationship with their groups, with children who were happy to meet such a dedicated figure every week and receive support from them. Thanks to the proactive support, children kept their attention longer and were guided step-by-step in finding the best solution for their envisioned video games (code: *interaction* between tutors and students).

Besides the positive aspects, we also coded different negative aspects that emerged during the course:

- **(–)** Most of the time, class teachers were not able to actively participate in the proposed activities due to a lack of skills in computing education. Even after following the training sessions before the start of the course, teachers could not understand how the basic concepts used for developing the video games, *e.g.*, loops and conditional statements, operated (code: *teachers involvement*).
- **(–)** Some high-school students demonstrated a general lack of interest in their tasks, often disclosing the solution to their groups at the beginning of the lesson. This prevented children from creatively exploring possibilities and solutions, undermining the envisioned trial-and-error approach (codes: *interaction* between tutors and students, *students' progress*).
- **(–)** Although most of the groups successfully cooperated in the project work, some groups were heterogeneous in terms of skills. This resulted in an unbalanced workload, with one child actively developing the video game and the other child passively attending the laboratory sessions (codes: *interaction* between students, *students' progress*).
- **(–)** Children with selective mutism disorder and Down syndrome in CLASS B had difficulties in fulfilling the project outcomes, even with the support of the teacher's aide (codes: *interaction* between teachers and students, *students' progress*).

Fig. 3. A group of children presenting their video game to the class. Compared to the example the expert showed at the beginning of the course, the group envisioned a football video game with points represented by goals.

Our findings align with prior research on supporting teachers in integrating CT into their curricula. Studies have emphasized the importance of structured guidance and active teacher involvement in fostering student engagement in computing education [43–45]. Our analysis further illustrates how a proactive, engaging, and supportive approach can help teachers navigate challenges in implementing CT, reinforcing previous work that highlights the need for pedagogical strategies tailored to primary education settings [46–48]. Moreover, by identifying patterns in student progress and teacher–student interactions, our study contributes to the broader discourse on effective CT learning environments, complementing existing frameworks on scaffolding and teacher facilitation [49,50].

5. Co-design study

Our observational study revealed that class teachers encountered difficulties in participating actively in the proposed activities, mainly due to a lack of skills in computing education, highlighting that the training activities we conducted before starting the course were not sufficient. We, therefore, recognized the value of a more collaborative approach between computing experts and class teachers in establishing how to integrate CT into students' computing education curricula. Consequently, we conducted a set of co-design activities with the four teachers who had participated in the observational study to assess the possibility of involving them to refine the first edition of the course by re-designing the included experimental activities. Participatory design research and its co-design approaches gained interest as a model for involving users and designers in the technology itself in a process of technological development [51]. These techniques, in fact, are innovative in the way they propose a shift in attitude from designing for users (*e.g.*, User-Centered Design) to one of designing with users. The final aim is to ensure the result meets the user groups' needs and requirements, developing an engaging design process that successfully involves all the interested actors, such as the stakeholders, designers, researchers, and end users. Our co-design efforts allowed us to consolidate our observations' implications into the educational model presented at the end of this paper (Section 6).

5.1. Methodology

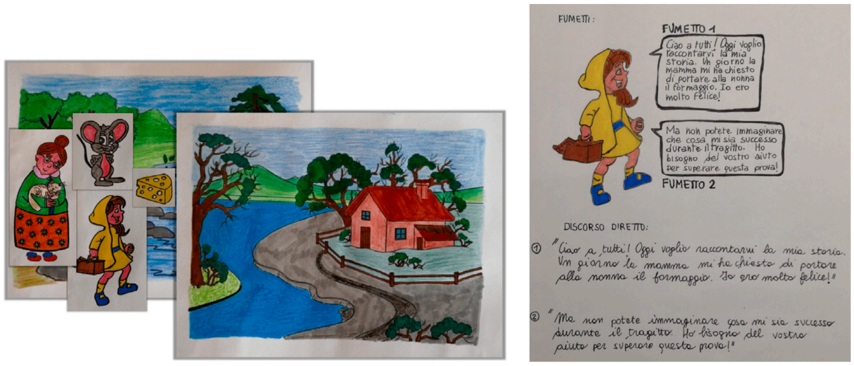
Co-design has been recently adopted as a strategy to facilitate the design and implementation of CT curricula [52,53]. To conduct the co-design phase of our study, we relied on a mutual learning approach [54] where we, as computing experts, strived to learn about the practices and contexts of the class teachers while they, in turn, strive to learn about possible technological options. This process underscored a synergistic effort between researchers and teachers, leveraging their complementary expertise to refine the first edition of the course presented in Section 3. Teachers contributed their pedagogical insights and deep understanding of their students' needs, while researchers provided technological expertise and guidance. This collaboration fostered a mutual learning environment where both groups enriched their knowledge and perspectives, particularly regarding the integration of CT into existing curricular frameworks. The co-design approach empowered teachers to actively shape the curriculum, enhancing their confidence and sense of ownership in delivering the content. At the same time, researchers collaboratively worked with teachers to address gaps in their technological skills, creating a curriculum tailored to both their professional growth and the students' engagement.

5.1.1. Participants and procedure

The four teachers involved in the observational study (see Section 4.1.1) participated in 6 sessions lasting 90 min each. During the course, two (T1 and T2) participated as class teachers, while the others (T3 and T4) were involved as teacher aides supporting children with disabilities and learning disorders.

The six meetings were conducted as follows:

- The first two sessions were conducted as a focus group to assess the needs, feelings, and challenges experienced by the teachers during the course and compare teachers' reflections with the results of our observations. One researcher moderated these two sessions, keeping the discussion on track without inhibiting the flow of thoughts and comments. Another researcher, instead, was in charge of taking notes.
- The remaining four sessions were instead structured as co-design activities. During these sessions, we challenged the teachers to collaboratively design and implement a new gamified project to be adopted in the course. The only constraints that teachers were given were the usage of the Scratch platform and the course's duration, *i.e.*, six 90-minute lessons. First, they had to design the project: in this phase, teachers brainstormed possible solutions and used papers and pens to define the students' expected outcomes. Then, they had to translate the envisioned project on the



(a) Drawing   (b) Storytelling

Fig. 4. In the first co-designed project, teachers envisioned backgrounds and characters (a) drawn by hands and the usage of a storytelling approach (b) through which students can consolidate grammar topics, *e.g.*, direct and reported speech.

Scratch platform. Two researchers supervised the four co-design sessions, helping teachers avoid unrealistic ideas and supporting them in implementing the project in Scratch.

5.2. Results

During the focus groups, teachers confirmed most of the results of our observational study. On the one hand, they said their students highly appreciated the course and demonstrated interest in the experimental activities outside the dedicated class hours. T1, for example, said *"one of my students always wanted to show me something, saying that he was developing other video games at home with the help of his father"*.

On the other hand, teachers also acknowledged that their lack of technological skills prevented them from actively interacting with the class during the course. All of them, in particular, said that the training activities conducted before the course were insufficient. Before starting the co-design activities, they explicitly asked us for a comprehensive review of the Scratch platform. Interestingly, T2 and T3 pointed out that co-designing a project for a new course's version is also the best way to learn how to implement it: *"if it's something that is tailored to my class and created by me...well, I would definitely remember it better"* (T2).

As observed in our previous study, teachers noted that the contribution of the high-school students was good in some circumstances and poor in others. Teachers highlighted consistent differences between high-school students of the involved classes, saying that some of them were highly motivated while others demonstrated very poor motivation and attitude. T4 suggested establishing a recruiting process at the beginning of the course, *e.g.*, through some questionnaires or by considering the students' outcomes, to recruit highly motivated students only. T1, instead, proposed to provide further incentives to the involved students, *e.g.*, with an evaluation given by the expert at the end of the course.

Finally, all the teachers agreed on the importance of having a stronger link between the project developed during the course and the topics normally covered throughout the year. T2, for example, said that *"using Scratch could be a good way to teach our topics, like math problems, under a different perspective – especially if we maintain this video game component"*.

Co-designed projects. The co-design activities resulted in two example projects.

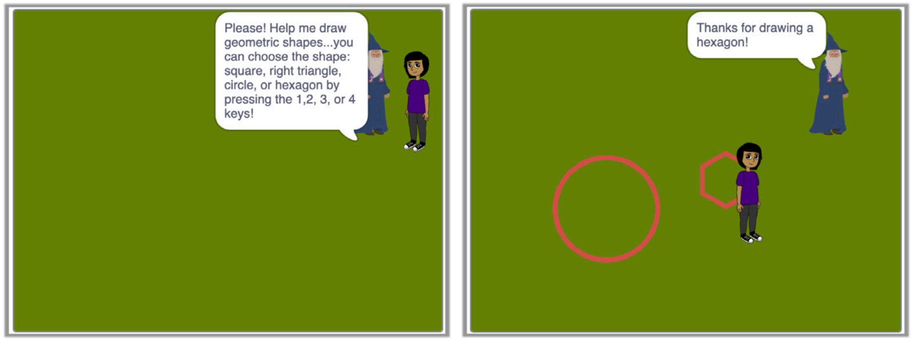
Fig. 4 reports some sketches produced by the teachers during a co-design session related to the first example project. The envisioned project focused on a storytelling and drawing approach, strongly linked with art and grammar subjects. Overall, the project is composed of two main phases:

Storytelling and Drawing. The teacher chooses a story by an author from those covered during the school year. After reading the story, the teacher asks the children's groups to identify and draw a set of scenes and characters on a sheet of paper, taking inspiration from the story told. For example, Fig. 4(a) shows some backgrounds and characters drawn by the teachers during the co-design sessions. Then, teachers assist children in defining a simple game involving the selected scenes and characters, similar to the one characterizing the first edition of the course. In the co-designed example reported in Fig. 4, teachers took inspiration from a revisitation of the Little Red Riding Hood fairy tale. They envisioned a game where a Little *Yellow* Riding Hood must collect as many cheese slices as possible for her grandmother before a mouse can reach them. Children are also asked to define simple dialogues between their characters to introduce the game (see Fig. 4(b), for an example): this is also a good opportunity to consolidate grammar concepts seen during the year, *e.g.*, direct and reported speech.

Game Implementation. With the help of experts, teachers, and high-school students, students import their drawings on Scratch, *e.g.*, by taking some pictures with their tutor's smartphone. Following the same educational strategies of the original course, children are challenged to implement the initial dialogues between their characters: in the co-designed project, for example, there is a dialogue between Little Yellow Riding Hood and her grandmother that introduces the fairy tale and defines the objectives of the game, *i.e.*, collecting cheese slices. Then, as in the original course, students can implement the game, exploring basic computing concepts like conditional statements and loops. This may involve implementing the movement of the main character, the (automatic) movement of the enemy, and the management of points (*i.e.*, collected cheese slices).

In a second co-designed project, teachers envisioned a programming course more linked to geometry. During the co-design activities, teachers said that most of their students struggled with basic geometrical concepts and tasks like calculating a square's area or identifying a triangle's angles. They all agreed that a good idea to address these problems was to develop a video game where the main character is asked to draw geometrical shapes by (automatically) moving on the screen.

Fig. 5 reports some screenshots of the second project developed on Scratch. As in the original course, teachers decided to have two characters interacting together to make students explore basic computing concepts, *e.g.*, conditional statements. When the main character approaches the wizard, the latter asks the character to draw a geometrical shape by clicking on a keyboard key (Fig. 5(a)). Students are



(a) Dialogues (b) Drawing

Fig. 5. A second example of a project envisioned by the teachers in a co-design session. In this case, teachers developed a Scratch program in which instructions are given by a character in the form of dialogues (a), and the main character is asked to draw geometric shapes (b).

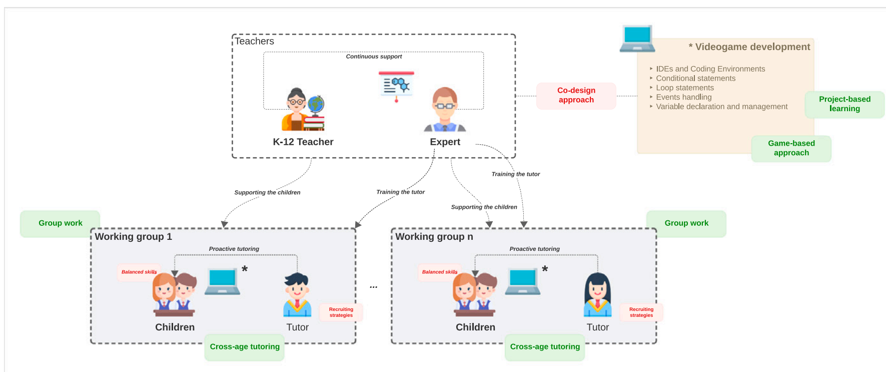


Fig. 6. A model to integrate computational thinking in K-12 education in an engaging, inclusive, and supportive way, focused on the collaboration between class teachers and computing experts.

therefore asked to associate the click of a keyboard key with a given shape, making the main character move and trace the path with a pen to replicate the shape (Fig. 5(b)). Shapes may vary depending on what the teacher has already done in geometry classes and may be easy or hard to draw. Drawing a square can be done with rectilinear movements of the same length and four 90-degree rotations. On the contrary, drawing a right-angled triangle requires students to define sides and acute angles, while drawing a circle can be made with 360 rotations of 1 degree.

6. A model for integrating computational thinking in K-12 computing education

Our work opens the way for empowering K-12 teachers to create their own educational pathway to integrate CT into their computing education curricula, creating courses that are customized to their class and involving multiple actors to support children proactively. In this section, we discuss the findings of our work, condensing them into an educational model.

The model is graphically represented in Fig. 6. We discuss it hereafter, linking its main parts to the literature and our findings. Overall, this model can be used by K-12 teachers to effectively integrate CT into their computing education curricula by collaborating with computing experts, with the final aim of creating tailored experiences that engage children in an inclusive and supportive way.

6.1. Educational strategies

The educational model presented in Fig. 6 encompasses three main educational strategies, *i.e.*, group work, project-based learning, and game-based approach, that can be mixed together to integrate CT into K-12 computing education engagingly and inclusively, thus minimizing the *curricula development* issue highlighted by the "Informatics for All" initiative [5]. We included and successfully tested these strategies from the beginning of our work (Fig. 1, A), with our evaluations that allowed us to refine them and understand how to better apply them in practice, especially for what concerns the group work. According to Burgstahler [55], who explored the application of Universal Design to the teaching domain, group work should reward the diverse students' contributions with respect to skills and roles. As reported in Section 4, our observations (Fig. 1, B) highlighted that some of the observed groups did not function properly, mainly because students had different skills. In these groups, one child actively developed the video game, while the other child passively attended the laboratory sessions as she could not keep up with her partner. As learning (especially learning by doing) is fundamental in primary education – much more than just finishing an exercise – our model specifies that groups should be composed of students with *balanced skills*, so that all of them can contribute equally.

Besides group work, our investigation suggests that project-based learning [31] – an approach that has been extensively used and explored for university courses [27,56–58] – may be an effective solution to introduce CT to primary school children. Although further studies are needed to confirm this finding, *e.g.*, by comparing it with alternative educational strategies, our observations (Fig. 1, B) and co-design activities with teachers (Fig. 1, C) outlined a promising way to engage challenging groups of kids in creating video games for their projects. The students highly appreciated such a gamification approach, enabling them to develop basic computing skills while fostering creativity. Gamification can be defined as "the use of game design elements in non-game contexts" (Deterding et al. [32]). Over the past few years, gamification has gained popularity in education and training due to its perceived capacity to enhance motivation and engagement in learning. However, the review of Caponetto et al. [59] highlights that gamification strategies are most often used at the university level. Particularly in *computing* education [60], gamification has been found to be an effective strategy for advanced topics like software testing [61] and machine learning [62]. Through our work, we have confirmed the need and effectiveness of using gamification strategies in primary schools [63] in the context of computing education. In addition, we have shown that such strategies can be adopted within a project-based learning approach in which kids develop their own video games. Overall, our model does not specify which specific game should be developed (this is something that is created through the collaboration between teachers and experts, see Section 6.3), but rather defines the main learning objectives, *i.e.*, IDEs and coding environments, conditional and loop statements, events handling, and variable declaration and management.

6.2. Involved actors in class

The model presented in Fig. 6 involves multiple actors, who collaborate to integrate CT in K-12 computing education and support students proactively. Such an approach helps create an engaging experience for children (*curricula development* issue) and, at the same time, minimize the *teacher training* issue, as teachers themselves receive support from experts. According to the model, children receive tutoring from three figures: the expert, the class teacher, and the high school tutor. Specifically, the model describes how such a tutoring approach should be implemented, explaining how the different actors interact in the classroom. While all of these figures support children during their group work, experts and teachers mainly interact with the entire class, *e.g.*, to introduce the experimental activities or to explain concepts and solutions at the end of the lesson. Furthermore, continuous, bidirectional support takes place between teachers and experts. With a deep knowledge of their students, teachers help experts interact with children in the best possible way. On the contrary, experts help teachers when they encounter difficulties explaining computing-related concepts. Instead, a "proactive tutoring" role is assigned to the groups' tutors, who guide children toward a possible solution. Tutoring and its trial-error dynamic comply with the Carl Rogers proposal [64], according to which the most meaningful learning is the one that translates into new behaviors. It happens when the teacher acts as a facilitator, creating an atmosphere where the subject can embark on a path of empowerment and skill development. The facilitating teacher stands in the wake of the educational culture oriented to the growth of the subjects' potential and activism-oriented methodologies, which place practical experience and the person at the center. Thus, theory acts through practice, placing abstract knowledge alongside forms that are as concrete as possible.

Our observational study (Fig. 1, B) and focus groups with teachers (Fig. 1, C) demonstrate that proactive support from high-school students is a valuable approach to be adopted for integrating CT into primary schools computing education curricula. The adopted tutoring strategy can be considered a particular form of cross-age tutoring [65]. In cross-age tutoring, some students take the role of tutors to help other students (typically younger) perform some class tasks. According to Cohen [66], such an approach can help eliminate the competition between peers that may hinder learning while enabling the tutor to engage in a meaningful interaction with younger students [67]. While an established practice, cross-age tutoring typically involves tutors and tutees belonging to the same educational level, *e.g.*, high-school students helping junior high-school students [67]. In our work, we explored a trans-generational form of cross-age tutoring, empowering high-school students (generation *Z*) to provide their support to primary school students (generation *Alpha*). We demonstrated in particular that high-school students who already have some experience with computing concepts can effectively help children develop simple video games in Scratch while exploring different solutions through a trial-error dynamic. At the same time, the presence of tutors who are closer in age to the primary school children can create a more relaxed and supportive environment and can encourage the children to ask for help without fear of being judged. Not surprisingly, our studies highlight that tutors' motivation is fundamental for such an approach to work properly. As suggested by the teachers in the focus groups, our educational model includes strategies for recruiting highly motivated students. These strategies may range from ad-hoc recruiting surveys to proper incentives.

6.3. Co-designing tailored experiences

One of the main implications of our work – that is related to *teacher training* issue – is that the involvement of class teachers is crucial in ensuring that computing education courses like the one described in this paper are integrated into the curriculum and aligned with the school's goals. Although the educational strategies included in the model were considered appropriate and effective by the teachers, the same teachers highlighted the need to exploit video games to explore topics usually covered throughout the year. While most current efforts to educate teachers about computing literacy have been mainly limited to computer science teachers [14], supporting other subject teachers in integrating CT into their subject areas and curricula is a critical aspect of promoting computing education [68]. In particular, research has shown that incorporating computing concepts into existing curricula can effectively allow students to apply computing skills to real-world problems and contexts [69]. In an effort to achieve such integration, Weintrop et al. [70] proposed a taxonomy of practices to integrate CT into high school math and science curricula. We envision the same approach to be implemented in primary schools.

As shown in Section 5, co-design – a well-known HCI method that has already been adopted in education [71] – might play a crucial role in ensuring that computing education courses meet the needs and interests of primary school students. Co-designing with teachers can help to ensure that the course is inclusive and accessible to a diverse range of students, *e.g.*, children with disabilities and learning disorders. At the same time, co-design activities involving experts and teachers may allow teachers to internalize better and learn computing concepts (*teachers training* issue) while ensuring that computing education courses for primary schools are adequately designed from a technical point of view. For these reasons, our model envisions co-designing activities involving teachers and experts to define specific video games, *e.g.*, related to other class subjects, to address the model's learning objectives. The projects envisioned by the teachers in the co-designed activities (Section 5), for example, contained strong links with grammar, art, and geometry subjects and demonstrated the feasibility of such an approach.

6.4. Limitations and future works

The model presented in this section has been rigorously defined by condensing and abstracting insights from observational and co-design studies conducted in Italy. While we anticipate the model's generalizability to countries with similar socio-economic statuses, further research is required to evaluate its applicability in other countries with potentially different levels of digital skills and technology usage. Overall, our focus has been on developing the model to provide primary school teachers with a systematic process to proactively, engagingly, and supportively integrate CT into their computing education curricula. In our upcoming work, we plan to apply the model in a large-scale case study to quantitatively assess its effectiveness in addressing the issues of curricula development and teacher training that undermine K-12 computing education.

7. Conclusions

In this paper, we have proposed an educational model to integrate CT in K-12 computing education in an engaging, inclusive, and supportive way, encompassing the collaboration between different actors, *i.e.*, K-12 teachers, computing experts, and high-school tutors. We defined the model stemming from the design and evaluation of an introductory coding course exploiting the Scratch programming platform. The course integrated group work, project-based learning, and gamification strategies. Furthermore, it implemented a cross-age tutoring approach through which high-school students proactively assisted children following a trial-error dynamic. An observational study and a set of co-design activities with the involved teachers demonstrated the effectiveness of the proposed educational and support strategies, allowing us to refine and generalize our choices into the educational model that can be used to minimize two well-acknowledged challenges characterizing the integration of CT in K-12 computing education, *i.e.*, the creation of tailored and engaging experiences for children and the need of supporting teachers without strong technological backgrounds.

8. Selection and participation of children

The study's participants were 4th-grade students, K-12 teachers, and high-school students. All of them were from public Italian schools. High-school students were recruited on a voluntary basis from a technical public institute, and their participation had the consensus of their teachers. The observational study took place in two 4th classes of a primary school in a large Italian city, in rooms strictly designated to the experimental setup. Data related to the study were collected after approval from our university and the public foundation that financed the "*Batti il 5*" project, following all the regulations and recommendations for research with children. A researcher contacted the school principal and legal guardian of each child to obtain written consent permitting the data and photo/screenshots collection.

CRediT authorship contribution statement

Alberto Monge Roffarello: Writing – original draft, Methodology, Investigation, Conceptualization. **Juan Pablo Sáenz:** Writing – review & editing, Writing – original draft, Methodology, Investigation, Conceptualization.

Declaration of competing interest

There are no conflicts of interest to be acknowledged

Acknowledgments

This work was carried out within the "Batti il cinque! Progettualità comunitarie tra nord e sud" project (2017-GEN-00335) – funded by "Impresa Sociale 'Con i bambini', soggetto attuatore del 'Fondo per il contrasto alla povertà educativa minorile' costituito dalla Legge n. 208/2015".

Data availability

No data was used for the research described in the article.

References

[1] I. Lee, F. Martin, J. Denner, B. Coulter, W. Allan, J. Erickson, J. Malyn-Smith, L. Werner, Computational thinking for youth in practice, ACM Inroads 2 (1) (2011) 32–37, http://dx.doi.org/10.1145/1929887.1929902.

[2] E. Commission, J.R. Centre, K. Engelhardt, Y. Punie, A. Chioccariello, A. Ferrari, G. Dettori, P. Kampylis, S. Bocconi, Developing Computational Thinking in Compulsory Education - Implications for policy and practice, Publications Office, 2016, http://dx.doi.org/10.2791/792158.

[3] S. Papavlasopoulou, M.N. Giannakos, L. Jaccheri, Exploring children's learning experience in constructionism-based coding activities through design-based research, Comput. Hum. Behav. 99 (2019) 415–427, http://dx.doi.org/10.1016/j.chb.2019.01.008.

[4] J.J. Lu, G.H. Fletcher, Thinking about computational thinking, in: Proceedings of the 40th ACM Technical Symposium on Computer Science Education, SIGCSE '09, Association for Computing Machinery, New York, NY, USA, 2009, pp. 260–264, http://dx.doi.org/10.1145/1508865.1508959.

[5] M.E. Caspersen, J. Gal-Ezer, A. McGettrick, E. Nardelli, Informatics for All The Strategy, Tech. rep., Association for Computing Machinery, New York, NY, USA, 2018, http://dx.doi.org/10.1145/3185594.

[6] G. Di Pietro, Changes in Italy's education-related digital divide, Econ. Aff. 41 (2) (2021) 252–270, http://dx.doi.org/10.1111/ecaf.12471.

[7] P. Mishra, A. Yadav, the Deep-Play Research Group, Rethinking technology & creativity in the 21st century, TechTrends 57 (3) (2013) 10–14, http://dx.doi.org/10.1007/s11528-013-0655-z.

[8] V. Barr, C. Stephenson, Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? ACM Inroads 2 (1) (2011) 48–54, http://dx.doi.org/10.1145/1929887.1929905.

[9] J.M. Wing, Computational thinking, Commun. ACM 49 (3) (2006) 33–35, http://dx.doi.org/10.1145/1118178.1118215.

[10] Ministero dell'Istruzione, dell'Università e della Ricerca, Buona scuola: la legge di riforma dell'istruzione, 2015, https://www.miur.gov.it/documents/20182/0/La+Buona+Scuola.pdf/b01ce9d1-9b1d-2f8a-e0bb-43430d16f83c.

[11] L. Gabriele, F. Bertacchini, P. Pantano, E. Bilotta, Laboratory for learning 21st century skills: training paths for pre-service primary school teachers using scratch, Ital. J. Educ. Technol. 28 (1) (2020) 20–42, http://dx.doi.org/10.17471/2499-4324/1113.

[12] G. Lombardi, The Role of Unplugged Coding Activity in Developing Computational Thinking in Ages 6-11, IGI Global, 2020, http://dx.doi.org/10.4018/978-1-6684-2411-7.ch015.

[13] I. Corradini, M. Lodi, E. Nardelli, Computational thinking in Italian schools: Quantitative data and teachers' sentiment analysis after two years of "programma il futuro", in: Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education, ITiCSE '17, Association for Computing Machinery, New York, NY, USA, 2017, pp. 224–229, http://dx.doi.org/10.1145/3059009.3059040.

[14] A. Yadav, C. Mayfield, N. Zhou, S. Hambrusch, J.T. Korb, Computational thinking in elementary and secondary teacher education, ACM Trans. Comput. Educ. 14 (1) (2014) http://dx.doi.org/10.1145/2576872.

[15] J. Simmonds, F.J. Gutierrez, C. Casanova, C. Sotomayor, N. Hitschfeld, A teacher workshop for introducing computational thinking in rural and vulnerable environments, in: Proceedings of the 50th ACM Technical Symposium on Computer Science Education, SIGCSE '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 1143–1149, http://dx.doi.org/10.1145/3287324.3287456.

[16] D.J. Ketelhut, K. Mills, E. Hestness, L. Cabrera, J. Plane, J.R. McGinnis, Teacher change following a professional development experience in integrating computational thinking into elementary science, J. Sci. Educ. Technol. 29 (1) (2020) 174–188, http://dx.doi.org/10.1007/s10956-019-09798-4.

[17] M.J. Muller, S. Kuhn, Participatory design, Commun. ACM 36 (6) (1993) 24–28, http://dx.doi.org/10.1145/153571.255960.

[18] F.J.J.M. Janssen, K.D. Könings, J.J.G. van Merriënboer, Participatory educational design: How to improve mutual learning and the quality and usability of the design? Eur. J. Educ. 52 (3) (2017) 268–279, http://dx.doi.org/10.1111/ejed.12229.

[19] J. Kelter, A. Peel, C. Bain, G. Anton, S. Dabholkar, M.S. Horn, U. Wilensky, Constructionist co-design: A dual approach to curriculum and professional development, Br. J. Educ. Technol. 52 (3) (2021) 1043–1059, http://dx.doi.org/10.1111/bjet.13084.

[20] M. Videnovik, T. Vold, L. Kiønig, A. Madevska Bogdanova, V. Trajkovik, Game-based learning in computer science education: a scoping literature review, Int. J. STEM Educ. 10 (1) (2023) 54, http://dx.doi.org/10.1186/s40594-023-00447-2.

[21] I. Morrison, T. Everton, J. Rudduck, J. Cannie, L. Strommen, Pupils helping other pupils with their learning: Cross-age tutoring in a primary and secondary school, Mentor. Tutor.: Partnersh. Learn. 8 (3) (2000) 187–200, http://dx.doi.org/10.1080/713685535.

[22] K.J. Topping, C. Buchs, D. Duran, H. van Keer, Effective peer learning: From principles to practical implementation, 2017.

[23] S. Tenhovirta, T. Korhonen, P. Seitamaa-Hakkarainen, K. Hakkarainen, Cross-age peer tutoring in a technology-enhanced STEAM project at a lower secondary school, Int. J. Technol. Des. Educ. 32 (3) (2022) 1701–1723, http://dx.doi.org/10.1007/s10798-021-09674-6.

[24] J. Moreno-León, G. Robles, Code to learn with scratch? A systematic literature review, in: 2016 IEEE Global Engineering Education Conference, EDUCON, 2016, pp. 150–156, http://dx.doi.org/10.1109/EDUCON.2016.7474546.

[25] H. Montiel, M.G. Gomez-Zermeño, Educational challenges for computational thinking in K–12 education: A systematic literature review of "scratch" as an innovative programming tool, Computers 10 (6) (2021) http://dx.doi.org/10.3390/computers10060069.

[26] I. Cetin, Preservice teachers' introduction to computing: Exploring utilization of scratch, J. Educ. Comput. Res. 54 (7) (2016) 997–1021, http://dx.doi.org/10.1177/0735633116642774.

[27] F. Corno, L. De Russis, Training engineers for the ambient intelligence challenge, IEEE Trans. Educ. 60 (2017) 40–49, http://dx.doi.org/10.1109/TE.2016.2608785.

[28] D.W. Johnson, R.T. Johnson, E.J. Holubec, Cooperation in the Classroom, Seventh ed., Interaction Book Company, 1993.

[29] E.H. Chiriac, Group work as an incentive for learning–students' experiences of group work, Front. Psychol. 5 (2014) 558, http://dx.doi.org/10.3389/fpsyg.2014.00558.

[30] D. Fung, Promoting critical thinking through effective group work: A teaching intervention for Hong Kong primary school students, Int. J. Educ. Res. 66 (2014) 45–62, http://dx.doi.org/10.1016/j.ijer.2014.02.002.

[31] L. Michaelsen, A. Knight, L. Fink, Team-Based Learning: a Transformative Use of Small Groups, 2002.

[32] S. Deterding, D. Dixon, R. Khaled, L. Nacke, From game design elements to gamefulness: Defining "gamification", in: Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments, MindTrek '11, Association for Computing Machinery, New York, NY, USA, 2011, pp. 9–15, http://dx.doi.org/10.1145/2181037.2181040.

[33] H.P. Young, Learning by trial and error, Games Econom. Behav. 65 (2) (2009) 626–643, http://dx.doi.org/10.1016/j.geb.2008.02.011.

[34] Y. Shenderovich, A. Thurston, S. Miller, Cross-age tutoring in kindergarten and elementary school settings: A systematic review and meta-analysis, Int. J. Educ. Res. 76 (2016) 190–210, http://dx.doi.org/10.1016/j.ijer.2015.03.007.

[35] J. Hollan, E. Hutchins, D. Kirsh, Distributed cognition: Toward a new foundation for human-computer interaction research, ACM Trans. Comput.- Hum. Interact. 7 (2) (2000) 174–196, http://dx.doi.org/10.1145/353485.353487.

[36] Fondazione della Comunità di Mirafiori Onlus, Batti il 5!, 2019, https://fondazionemirafiori.it/batti-il-5-progetto.

[37] J.B. Freeman, A.M. Garcia, L.M. Miller, S.P. Dow, H.L. Leonard, Selective mutism, in: T.L. Morris, J.S. March (Eds.), Anxiety Disorders in Children and Adolescents, The Guilford Press, 2004, pp. 280–301.

[38] S.Z. Ramirez, V. Morgan, Down syndrome, in: L. Phelps (Ed.), Health-Related Disorders in Children and Adolescents: A Guidebook for Understanding and Educating, American Psychological Association, 1998, pp. 254–265, http://dx.doi.org/10.1037/10300-036.

[39] J.G. Adair, The hawthorne effect: A reconsideration of the methodological artifact, J. Appl. Psychol. 69 (2) (1984) 334–345, http://dx.doi.org/10.1037/0021-9010.69.2.334.

[40] L.S. Nowell, J.M. Norris, D.E. White, N.J. Moules, Thematic analysis: Striving to meet the trustworthiness criteria, Int. J. Qual. Methods 16 (1) (2017) 1609406917733847, http://dx.doi.org/10.1177/1609406917733847.

[41] R. Rampin, V. Steeves, S. DeMott, Taguette: open-source qualitative data analysis, 2021, http://dx.doi.org/10.5281/zenodo.5111814.

[42] M.S. Linneberg, S. Korsgaard, Coding qualitative data: A synthesis guiding the novice, Qual. Res. J. 19 (3) (2019) 259–270.

[43] H. Killen, M. Coenraad, V. Byrne, L. Cabrera, K. Mills, D.J. Ketelhut, J.D. Plane, Teacher education to integrate computational thinking into elementary science: A design-based research study, ACM Trans. Comput. Educ. 23 (4) (2023) http://dx.doi.org/10.1145/3618115.

[44] C. Kim, B.R. Belland, L. Vasconcelos, R.B. Hill, Playful programming, social resilience, and persistent actions as drivers of preservice early childhood teachers' engagement in computer science, Sage Open 14 (4) (2024) 21582440241284488, http://dx.doi.org/10.1177/21582440241284488.

[45] N. Karlsen, E.K. Henriksen, K. Pajchel, Assessing teachers' knowledge of how to use computer programming in science and technology education, J. Sci. Educ. Technol. (2024) http://dx.doi.org/10.1007/s10956-024-10145-5.

[46] E. Polat, Z. Kadirhan, S. Hopcan, H. Yildiz Durak, S. Canbazoglu Bilici, Teaching motivation of in-service teachers, professional identity, and self-efficacy for teaching computational thinking, Eur. J. Educ. 60 (1) (2025) e12928, http://dx.doi.org/10.1111/ejed.12928, e12928 EJED-2024-0950.R2.

[47] N. Bean, J. Weese, R. Feldhausen, R.S. Bell, Starting from scratch: Developing a pre-service teacher training program in computational thinking, in: 2015 IEEE Frontiers in Education Conference, FIE, 2015, pp. 1–8, http://dx.doi.org/10.1109/FIE.2015.7344237.

[48] A. Yadav, N. Zhou, C. Mayfield, S. Hambrusch, J.T. Korb, Introducing computational thinking in education courses, in: Proceedings of the 42nd ACM Technical Symposium on Computer Science Education, SIGCSE '11, Association for Computing Machinery, New York, NY, USA, 2011, pp. 465–470, http://dx.doi.org/10.1145/1953163.1953297.

[49] L. Cabrera, D.J. Ketelhut, K. Mills, H. Killen, M. Coenraad, V.L. Byrne, J.D. Plane, Designing a framework for teachers' integration of computational thinking into elementary science, J. Res. Sci. Teach. 61 (6) (2024) 1326–1361, http://dx.doi.org/10.1002/tea.21888.

[50] A. Yadav, C. Stephenson, H. Hong, Computational thinking for teacher education, Commun. ACM 60 (4) (2017) 55–62, http://dx.doi.org/10.1145/2994591.

[51] P.M. Asaro, Transforming society by transforming technology: the science and politics of participatory design, Account. Manag. Inf. Technol. 10 (4) (2000) 257–290, http://dx.doi.org/10.1016/S0959-8022(00)00004-7.

[52] F.J. Agbo, S.S. Oyelere, J. Suhonen, et al., Co-design of mini games for learning computational thinking in an online environment, Educ. Inf. Technol. 26 (2021) 5815–5849, http://dx.doi.org/10.1007/s10639-021-10515-1.

[53] A.O. Sunday, Design and implementation of co-design pedagogical scenarios for learning computational thinking, in: Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 2, in: ITiCSE 2023, Association for Computing Machinery, New York, NY, USA, 2023, pp. 609–610, http://dx.doi.org/10.1145/3587103.3594139.

[54] T. Robertson, J. Simonsen, Participatory design: An introduction, in: Routledge International Handbook of Participatory Design, Routledge, 2012, pp. 1–18.

[55] S. Burgstahler, Universal design: Implications for computing education, ACM Trans. Comput. Educ. 11 (3) (2011) http://dx.doi.org/10.1145/2037276.2037283.

[56] N. Hosseinzadeh, M.R. Hesamzadeh, Application of project-based learning (PBL) to the teaching of electrical power systems engineering, Educ. IEEE Trans. on 55 (2012) 495–501, http://dx.doi.org/10.1109/TE.2012.2191588.

[57] R. Rengel, M. Martin, B. Vasallo, Supervised coursework as a way of improving motivation in the learning of digital electronics, Educ. IEEE Trans. on 55 (2012) 525–528, http://dx.doi.org/10.1109/TE.2012.2194293.

[58] G. Gonçalves, S. Nicola, L. Afonso, M. Ferreira, T. Ferro, J. Mendonça, Problem based learning: An experience in computer engineering, J. Educ. Hum. Dev. 3 (2014) http://dx.doi.org/10.15640/jehd.v3n4a24.

[59] I. Caponetto, J. Earp, M. Ott, Gamification and education: a literature review, in: Proceedings of the 8th European Conference on Games-Based Learning - ECGBL 2014, vol. 1, 2014, pp. 50–57.

[60] Z. Zhan, L. He, Y. Tong, X. Liang, S. Guo, X. Lan, The effectiveness of gamification in programming education: Evidence from a meta-analysis, Comput. Educ.: Artif. Intell. 3 (2022) 100096, http://dx.doi.org/10.1016/j.caeai.2022.100096.

[61] T. Fulcini, R. Coppola, L. Ardito, M. Torchiano, A review on tools, mechanics, benefits, and challenges of gamified software testing, ACM Comput. Surv. (2023) http://dx.doi.org/10.1145/3582273, Just Accepted.

[62] A. Khakpour, R. Colomo-Palacios, Convergence of gamification and machine learning: A systematic literature review, Technol. Knowl. Learn. 26 (2021) 597–636, http://dx.doi.org/10.1007/s10758-020-09456-4.

[63] K. Nand, N. Baghaei, J. Casey, B. Barmada, F. Mehdipour, H.-N. Liang, Engaging children with educational content via gamification, Smart Learn. Environ. 6 (1) (2019) 6, http://dx.doi.org/10.1186/s40561-019-0085-2.

[64] C.R. Rogers, Client-Centered Therapy: Its Current Practice, Implications and Theory, Houghton Mifflin, 1951.

[65] J. Gaustad, Peer and cross-age tutoring, in: Striving for Excellence: The National Education Goals 2, National Education Goals Panel, 1993, p. 27.

[66] J. Cohen, Theoretical considerations of peer tutoring, Psychol. Sch. 23 (2) (1986) 175–186.

[67] A. Yogev, R. Ronen, Cross-age tutoring: Effects on tutors' attributes, J. Educ. Res. 75 (4) (1982) 261–268.

[68] J. Voogt, P. Fisser, J. Good, P. Mishra, A. Yadav, Computational thinking in compulsory education: Towards an agenda for research and practice, Educ. Inf. Technol. 20 (4) (2015) 715–728, http://dx.doi.org/10.1007/s10639-015-9412-6.

[69] M. Coenraad, L. Cabrera, H. Killen, J. Plane, D.J. Ketelhut, Computational thinking integration in elementary teachers' science lesson plans, in: Computational Thinking in PreK-5: Empirical Evidence for Integration and Future Directions, Association for Computing Machinery, New York, NY, USA, 2022, pp. 11–18.

[70] D. Weintrop, E. Beheshti, M. Horn, K. Orton, K. Jona, L. Trouille, U. Wilensky, Defining computational thinking for mathematics and science classrooms, J. Sci. Educ. Technol. 25 (1) (2016) 127–147, http://dx.doi.org/10.1007/s10956-015-9581-5.

[71] R. Nicholson, T. Bartindale, A. Kharrufa, D. Kirk, C. Walker-Gleaves, Participatory design goes to school: Co-teaching as a form of co-design for educational technology, in: Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, CHI '22, Association for Computing Machinery, New York, NY, USA, 2022, http://dx.doi.org/10.1145/3491102.3517667.