robotics

MDPI

Communication

Computational Design Thinking and Physical Computing: Preliminary Observations of a Pilot Study

Dochshanov Alden * and Michela Tramonti

European Training and Research Association for a Cooperation Key to business (EU-Track), 04019 Terracina, Italy; m.tramonti@eu-track.eu

* Correspondence: a.dochshanov@eu-track.eu; Tel.: +39-0773-721-606

Received: 20 July 2020; Accepted: 9 September 2020; Published: 10 September 2020

Abstract: Today's technological development inevitably defies educational approaches in terms of future demand for skills to be imparted. Among other skills, the capacity to operate and communicate effectively within multidisciplinary realms is duly considered as the fundamental one. Educational robotics (ER) and STEM do constitute a suitable framework for the development of these specific skills. Moreover, competences such as computational (CT) and design thinking (DT) have already been nominated as necessary to adapt to the future and relevant for innovation. The years of independent development and evidence of practical implementation justify the maturity of the related methodological approaches and emerging gradual shift towards their combination. In this regard, the actual work presents a pilot experience of the combined application of computational design thinking and educational robotics in the case of a 9-to-11-year-old target audience. The approach utilizes a novel platform developed under the project Coding4Girls combining design thinking and game-based learning and introduces physical computing through consecutive assembling and programming an IR-controlled robot-car. The core of the learning path consists in the development of primary programming skills and their gradual transfer into the physical realm. The method, as the study demonstrates, is capable of helping keep students both motivated and result-oriented throughout the duration of the course.

Keywords: computational thinking; design thinking; STEM; physical computing; robotics education; coding; multidisciplinarity

1. Introduction

Over the last few years, European schools have widely recognized computational thinking (CT) as one of the fundamental skills for the twenty-first century [1], thus favoring that coding be taught starting from primary school [2]. Stemming back from the constructionist work of Seymour Papert [3,4] and formulated and re-defined by Jeanette Wing [5,6], today the generalized definition of CT is synthesized as "the conceptual foundation required to solve problems effectively and efficiently (i.e., algorithmically, with or without the assistance of computers) with solutions that are reusable in different contexts" [7]. Coding, being closely adjacent to CT, concretizes its concepts and turns them into a tool for effective learning, due to promoting the abstractions of the real-world problem by designing and reasoning on computation artefacts such as programs [2]. From the other side, design thinking (DT) [8,9], representing a paradigm for dealing with problems in many professions [10], has matured into the approach for creative problem solving [11] that focuses on multidisciplinary teams and the involvement of all perspectives [12–14]. In education, sometimes referred to as "design-based learning" (DBL) [15], the method is perceived as "a model for enhancing creativity, endurance, engagement

and innovation" [16]. When combined with CT methodology, DBL has recently been shown to empower the impact of the former [17–19], despite being less frequently adopted in general [20]. Similarly, educational robotics (ER) with its deep roots in problem-solving [21–24], collaborative learning, and engineering mindset cultivation [25–27] has also been universally acknowledged to favor and advance CT skills development [28–34].

Nevertheless, despite being closely related and mutually beneficial, e.g., in emphasizing various CT components [7], the methods have mainly been considered in their different pairwise combinations [30,35–40], but rarely complete [31,41–43]. For example, the study of Leonard et al. [31,42] looked at the potential of engaging youth in three areas: (a) robotics, (b) digital gaming, and (c) computational thinking through the use of LEGO® robotics kits, MINDSTORMS® software (LEGO Group, Billund, Denmark), and AgentSheets/AgentCubes object-based programming environments used for game design. Among other significant results, upon the constructs of videogaming, computer gaming, and computer use by type of treatment (gaming only or robotics/gaming) examination, the authors revealed that students in the robotics/gaming clubs had significantly higher scores on video gaming. They rightfully believe that "the excitement of getting a robot to move and perform a task combined with the creativity associated with developing a game from scratch influenced this result". Whereas the work of Valls et al. [41], through the use of Scratch and LEGO MINDSTORMS®, has demonstrated creativity to be a fundamental pillar in STEAM teaching. In particular, the final results revealed that contextualized activities, art, motivation, and collaboration enabled the control group to generate more creative, complex, and varying solutions and obtain better results in the areas of competence and knowledge of technology platform in a shorter time. Along with this, it is worth noting that in the works cited, the creative element has been considered as an essential, albeit auxiliary tool. In their work, Cross et al. [43] synthesized frameworks to identify talents in CT and Engineering Design (ED) through the originally developed Arts & Bots program [40,44] combining robotics components, craft materials, a custom programming environment, and transdisciplinary curricula to bring creativity-oriented technology experiences to students. By training non-technical teachers to be aware of computer science and engineering component skills, the authors reasonably believe that the diversity of future STEM innovators can be improved, which is perfectly in line with the actual needs outlined in [45]. The authors underline that despite being highly relevant to modern society, CT and ED are still out of the formal curricula. Besides, Ehsan et al. [46] add that there is still little research exploring how children's engagement in both engineering and computational thinking can support each other.

Moreover, as mentioned in [47], the expanding variety of new devices, from smartphones and tablets to electronic learning toys poses a steady challenge in defining developmentally appropriate activities and content for children of different ages; however, when a suitable curriculum is adopted children can be interested and able to learn many aspects of robotics, programming, and computational thinking.

In view of the above, this work aims at complementing existing literature on combined learning experiences unifying CT, DBL, and ER. The core of the approach presented consists in the development of the primary coding skills and concepts mastery with their subsequent transfer into the physical realm. While the first part utilizes the platform developed within the framework of the Coding4Girls project, the second represents the set of activities aimed at the assembly and programming of an IR-controlled robot car.

The article is organized as follows. First, the research design is outlined. Then the methodologies adopted for each phase, including the description of the platform developed under the Coding4Girls and robotics activities are presented. Finally, the results achieved throughout the work are given and analyzed.

2. Research Design

With a focus on verifying the effectiveness of CT, DBL, and ER combination, the present work was intended as an exploratory research for evaluating the preliminary feedbacks of the target audience to the proposed learning path. The activities were structured into a couple of consecutive sections. The first one was aimed at the introduction to the basic concepts of coding through the platform developed within the framework of Coding4Girls project. Subsequently, the skills mastered were deployed further in the second section, aimed at encouraging participants to deal with more complex tasks through assembling and programming an IR-controlled robot car.

A small group of participants (10 students aged 9 to 11) was considered, to enable more effective supervision of the learning environment and also merely due to the limited availability of hosting spaces and equipment (laptops, tablets, Arduino kits, etc.). The participation call was open for both genders. However, the only registered participants were the boys. Also, given the focus outlined and since the participation was voluntary, the variable "gender" was not considered as relevant at this stage.

Being extracurricular, the direct involvement of children in activities could be easily managed without the authorization of the teachers' council or the headmaster. Therefore, the authors, being the only ones participating in the teaching activities, were able to collect qualitative data and determine the learning context for a further in-depth study.

The training events were originally planned in the form of 25 informal meetings for a total of 50 h organized directly at the headquarters of EU-Track. However, in consequence of the critical situation that occurred due to the COVID-19 epidemic, face-to-face activities were not easy to manage. The corresponding measures taken are outlined in Section 4.1.

Thus, the pilot study aimed to answer the following general questions:

- How the planned activities will be perceived by the audience?
- What kinds of instruments (software/hardware) attract more attention?
- How may the skills deployment during the transfer between virtual and physical realms be assisted and optimized?

3. Methodology

The methodology of the learning path was meant to support the continuity of the skills to be mastered starting from the very beginning with their subsequent transfer into the physical realm. Below, the detailed presentation of the instruments adopted for both phases, namely "Coding and computational skills development" and "Educational robotics", is given. The participation of children in both phases mainly implied an individual involvement unless the particular activity required teamwork. In the last case, the students were divided in two subgroups of 5 members in each. The formation of teams was not based on any limiting criteria, thus pursuing the stimulation of communication skills and principles of self-organization.

3.1. Coding and Computational Skills Development

The first part of the designed learning path aimed to develop coding and computational thinking skills through game-based learning and design thinking approach. To do that, the Coding4Girls project methodology and tools were used [19]. This project was realized with the support of European Commission under the Erasmus+ Program and coordinated by the University of Ljubljana (Slovenia); and intends, mainly, to overcome the existing gap between male and female participation in computer science education and careers by introducing early methodological learning interventions that make computer science attractive to everyone.

The theoretical framework, upon which the new proposed pedagogical methodology has been designed, is a combination of design thinking and game-based learning approaches.

The design thinking approach is a creative process that helps students to design meaningful solutions collaboratively with their peers. Therefore, the design process is highly effective, thanks to its structured framework for identifying challenges, gathering information, generating potential solutions, refining ideas, and testing solutions. The process is recursive by nature and demands interaction. Each stage in the process is revisiting and invoking to encourage experimentation, solution feasibility, and reflection throughout a learning experience [15]. Thus, the related activities enable highly collaborative experience in and outside the classroom. Students are directly engaged in information gathering, knowledge generation, communication, and presentation.

In the Coding4Girls project, the designed thinking approach exploits the main advantages of game-based learning to encourage and motivate students in their learning processes. It uses some important game elements (e.g., points and challenges) and scenarios which make a certain activity more fun by increasing the degree of involvement, motivation, and results achieved.

In particular, the use of the challenges while helping students to see the big picture before designing a detailed solution encourages and challenges them to think entrepreneurially about digital technologies and how they can be used to address real-world problems.

On the base of this approach, the project team has developed the software consisting of two different interconnected parts. The first one is the Teachers' Platform and the second is the Students' Game Environment [48].

The students are encouraged to design and code games that address specific needs or issues (depending on teachers' choice). It is a "low entry high ceiling approach" allowing students to start with easy problems until engaging more challenging tasks. Teachers design and present to their students "half-baked" scenarios in which a solution is partially ready by challenging them to complete the task. Therefore, teachers can prepare small and manageable modules by using Coding4Girls (C4G) software suitable for their students' needs and knowledge level. The Snap! scenarios originally developed by the partnership and published in the Teachers' Platform and the Students' Game Environment were aimed more at girls than boys, but both the genders can benefit from the tools and the approach proposed.

Following the Design Thinking approach, the project team prepared some courses and learning scenarios designed to challenge students on solving a specific coding issue. The scenarios have different levels from basic to advance for more capable students. The tasks can be solved individually and collaboratively by arousing group discussion in the classroom or, virtually, on the Students' Game Environment. The software offers a brainstorming space where students can discuss and share their ideas by placing and managing multimedia post-it, as shown in Figure 1. All the students involved in the course can write and post their contribution which is visible to everyone enrolled in that specific course.

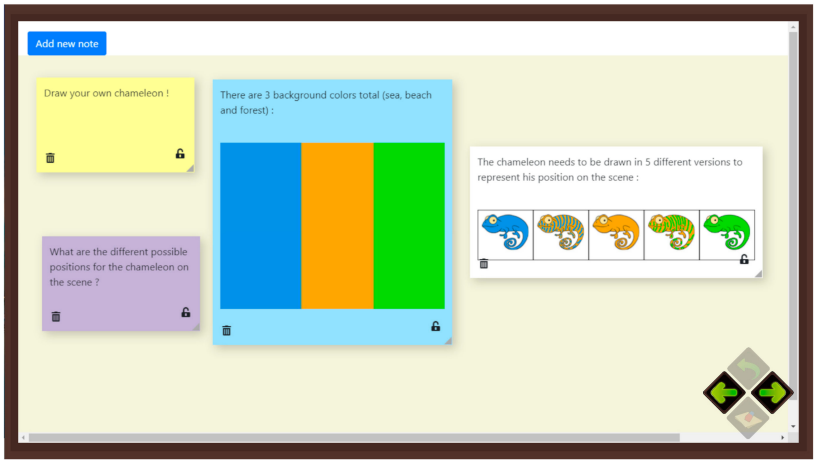


Figure 1. Brainstorm zone in the Students' Game Environment for students' contribution and discussion [19].

These specific coding courses are designed and prepared by the teachers by using Snap! canvas and published in the Teachers' Platform, a web-based platform (Figure 2).

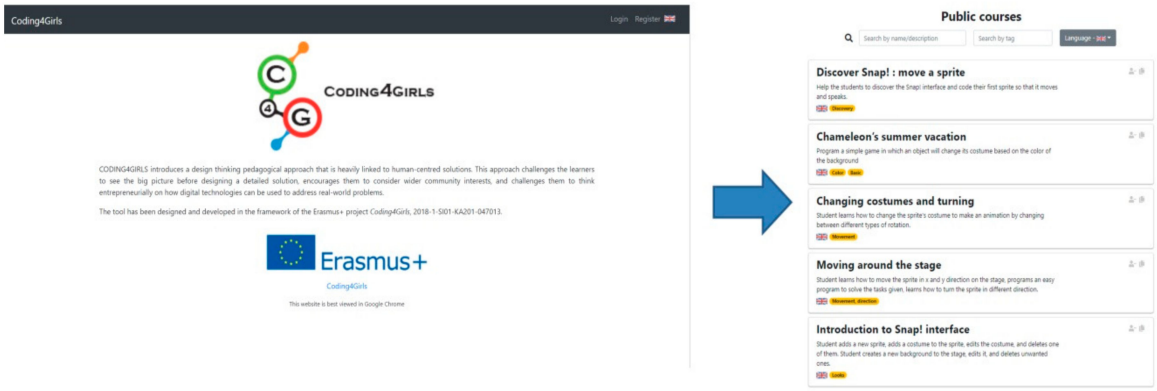


Figure 2. Teachers' Platform and some coding courses available in it.

These courses are constituted as a grouping space for thematically related activities, all connected to an overarching issue. The problem is presented at the very beginning of the course to the students who can brainstorm together to collaboratively elaborate tentative solutions. They can use post-its on a virtual board to discuss their ideas by using text, images or video (Figure 1). After the brainstorming phase, the students are given, in a step-by-step fashion, specific activities (presented in consecutive order) designed to present the tools necessary to solve the overarching problem. These coding activities will be presented by using Snap! canvas, through both the half-baked tasks to challenge the students in the problem solution and the final solution presentation.

In Figure 3, the structure of each C4G course published in the Teachers' Platform is shown.

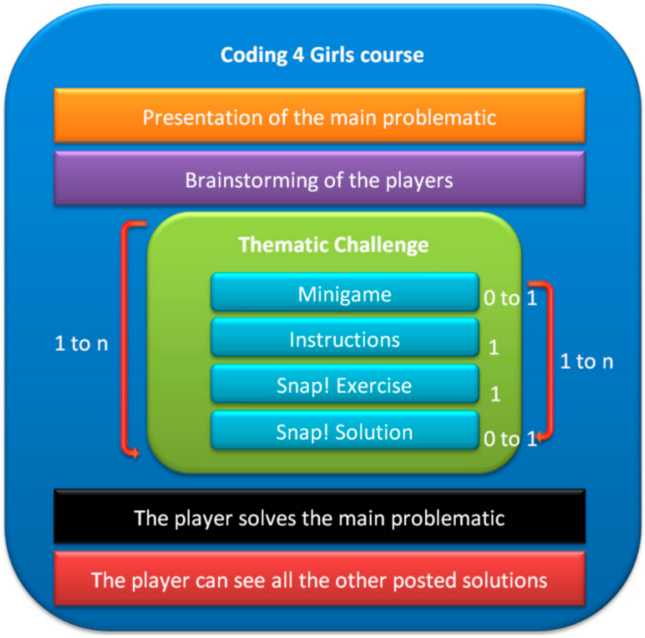


Figure 3. Structure of a Coding4Girls (C4G) course [19].

The grouped activities in the course are called challenges, which are presented to students in a specific order set by the teacher (Figure 4). For example, if a teacher wants to create a course on basic programming knowledge, the first activity could concern the concept of Booleans, the second could concern conditional structures and the last one, the loops. This will allow teachers to prepare

customized learning steps for their students in the Teachers' Platform and the students to be led gradually to the final learning objective in the Students' Game Environment.

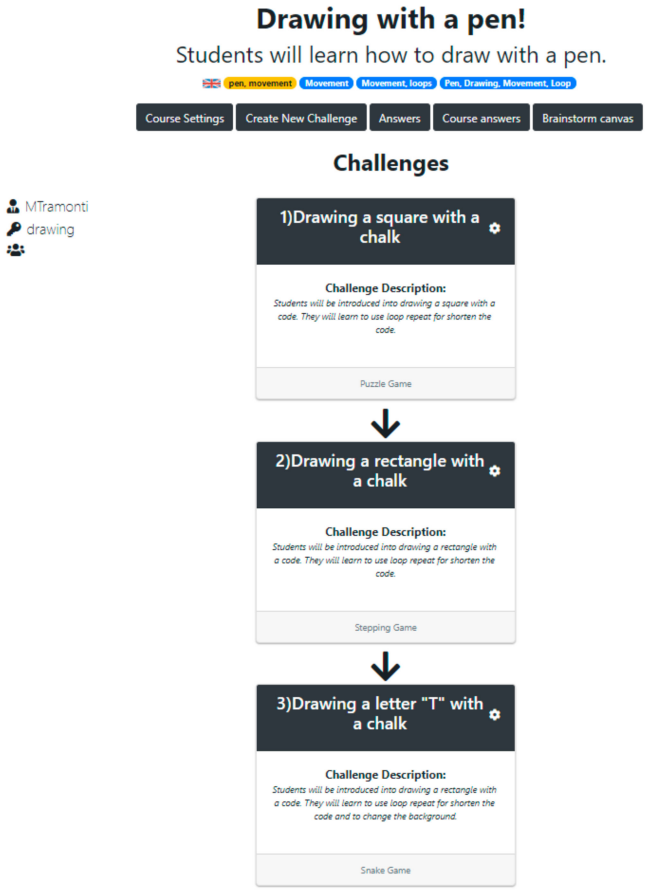


Figure 4. An example of challenges in a coding course in the Teachers' Platform.

Of course, students need to play and solve all the challenges in the order established by the teachers before achieving the final solution.

Each challenge might be associated with a mini game developed by the project team. The aim is to help the students understand better what could be the result of a specific coding property studied.

For example, if a challenge is concerned with the study of "loop" property, Match-3 is one of the mini games developed that could be associated with it. Therefore, each challenge might have, besides the Snap! canvas to solve the task, a mini game that the student will need to play following a page with instructions defined by the teacher during their preparation in the Teachers' Platform. However, it is not mandatory to include a mini game in the challenge.

The students will access these challenges through the Students' Game Environment that is a Unity 3D videogame downloadable software where students can discover and complete the courses prepared by their teachers in a fun, engaging, and playful manner.

All the games offered (currently 11) to the students are related to and simplify the actual programming concepts upon which the C4G courses and scenarios have been designed.

The courses, using elements of the design thinking approach, present to the students an overarching issue to solve and present the tools to solve it in a step-by-step approach. The courses as mentioned above, are prepared by teachers and published in the Teachers' Platform, but students can access them through the Students' Game Environment. Figure 5 shows the panel of challenges in the Students' Game Environment of the course created by the teacher in the Teachers' Platform. The number from 0 to 13 represents the challenges. When selecting one of them, the topic to be studied in the selected

challenge appears; for example, in our case is the "Conditional challenge". Besides the challenge topic, the system shows the 3D mini game associated with it, e.g., "Find your path".

Figure 5. The panel of challenges in the Students' Game Environment [19].

In the Students' Game Environment, each challenge is structured as follows: one introductory minigame illustrating the coding concept; one instructional page for the task to be fulfilled in Snap!; two Snap! canvases—one to be used to solve the task and the other to display the activity solution.

The teachers can decide which mini-game (optional) their students will play by selecting one of the existing minigames, such as Match3 game, Dice game (Figure 6), Inventory game, Find your path game, Stepping game, Sound game, Snake game, Puzzle game, Pattern matching game, and a multiple-choice question quiz.

Figure 6. An example of available mini-game - Dice Game [19].

The instructional page is in HTML, usually enriched by images or videos, to present the context and specific aim of the learning task to be fulfilled in Snap!

This page will be followed by a Snap! canvas, based on a template provided by the teacher containing the coding activity or problem to be executed or solved by students. Another Snap! canvas will be provided to display the solution of the challenge proposed. However, only the teacher can

decide if solution canvas is shown or not. It will depend on the teaching process management selected by the instructor.

Since every course assembles more challenges, these steps will be repeated as many times as the number of the total challenges identified by the teachers. This allows breaking one complex coding activity into simple elementary steps where the answer and/or the solution to the preceding activity becomes the template in which the next activity is to be executed.

In the end, the course is constituted of a certain number of challenges which present the programming teaching through an incremental scaffolding process.

Once the students have completed all the challenges of the course (Figure 7), they are led back to the initial coding problem and asked to synthesize the new knowledge they just acquired. At the very end of the course, the players can see also all the solutions to the problem proposed by the other students by arousing confrontation moments.

Figure 7. Students working with learning scenarios of the C4G platform.

In the framework of the Coding4Girls project, specific validation activities are foreseen and addressed to teachers from primary and secondary school with 10-to-16-year-old students coming from Slovenia, Bulgaria, Croatia, Greece, Italy, Portugal, and Turkey. In this context, the authors decided to use the project approach and tools to let the young participants have the first-hand experience with coding tasks, before starting with a more complex environment of programming, which Arduino IDE (Arduino S.r.l., Monza, Italy) can be. The details of the data analysis methodology are covered below in Section 3.3.

Although some of the pupils were at their first experience with coding, the methodology and tools provided were revealed to be a good means to let them understand and familiarize themselves with the main principles of programming. Therefore, at the end of the initial phase, the skills developed through personalized training activities allowed children to deal better with the second part addressed to introduce physical computing ideas, instruments, and reasoning.

3.2. Educational Robotics

This part of the learning path aimed at the further deployment of skills mastered previously with their gradual complementing by the hardware related issues. The variety of educational activities at this stage of the course can be divided into three basic types: background activities, introductory microcontroller programming, and direct work on the project using Arduino IDE. Following the paradigm shift in educational robotics and STEAM specified in [49,50], this part of the learning path aimed at encouraging learners to become makers of their own "transparent

robotics artefacts" [49]. Below, the methodology adopted in background activities and introductory microcontroller programming are given. Whereas the final part, related to the robot assembly and programming, is covered in the Results section.

3.2.1. Background Activities

The methodology adopted in background activities is as follows (Figure 8). Firstly, the corresponding physical phenomenon was introduced through a live demonstration. Afterwards, the related physical artefacts were shown and discussed. In particular, the audience was encouraged to provide different examples and situations where the components in question were met previously, thus leading towards the idea of the functionality of the elements. Once the basic functionality was formulated, to enable students' further grasping of the underlying concept deeper, they practiced playing with the argument both in real and virtual fashion.

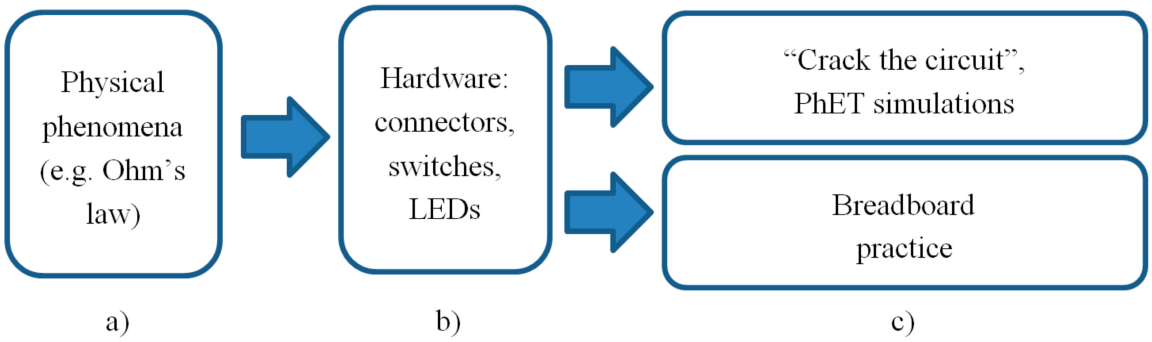


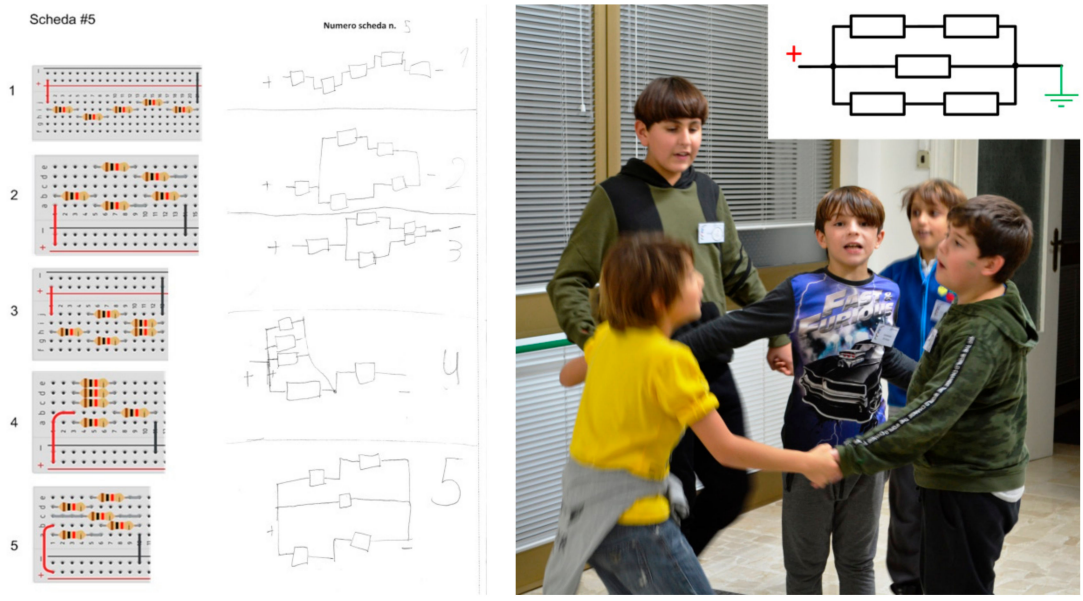
Figure 8. An example workflow of the educational robotics (ER) classes: (**a**) physical phenomena introduction; (**b**) related artefacts discussion; (**c**) playing with a concept virtually and in reality.

For example, playing around basic circuitry, the students used "Crack the circuit" [51], PhET Colorado University interactive simulations [52], and a customized breadboard. In such a way, given the concepts treated are seen and played with both virtually and physically, according to the authors' opinion, enables children to develop an expanded view of the artefacts' behavior. This, in turn, is perfectly in line with the microcontroller mindset initiation.

The background activities aimed to provide the primary acquaintance with the fundamental concepts of current, tension, resistance, and electric current power. Their units of measure, decimal prefixes, and the values are supposed to be dealt with while working on the final project. Without the concepts adequately grasped, any further movement may be superficial and therefore inadequate.

IR control, which was supposed to be introduced in the narrow sense of the concept during the last workshops of the course, at the beginning was nominated concerning the electromagnetic spectrum. The development of the concept demonstration involved a wide range of wave phenomena and harmonic motion. Starting from the simplest kind of harmonic motion of the pendulum, and students were accompanied to the hearable range of sound waves passing through several experiments using a couple of tuning forks demonstrations including Chladni plate [53] (using the function generator app [54] as the adjustable sound frequency source). From this perspective, as we have seen it, the students acquired a multi-perspective view of the phenomena in their different forms and physical manifestations.

Besides helping the students become accustomed to the interpretation of the breadboard circuitry (showing different combinations of resistors assembly) with the corresponding schematics representation (Figure 9a), the game-based activities foster the understanding of complex concepts for their age. In particular, there were several combinations of 10 resistors connected in series and parallel to be reproduced in a physical arrangement by the group of students (Figure 9b).



(**a**) (**b**)

Figure 9. Working around resistor networks: (**a**) the "from breadboard to schematics" challenge with an example of the corresponding solution proposed; (**b**) "play-time", building the resistor networks with the corresponding schematic challenge on the top-right.

3.2.2. Introductory Microcontroller Programming

To start imposing the "microcontroller mindset", during the introductory microcontroller programming activity, the micro:bit [55] platform was used. Due to its simplicity and visual "Scratch-like" online development environment, it has been found to be simple and effective in terms of further developing the microcontroller mindset. While going through several entry-level mini-projects [55], the students got accustomed to the association of the coding logics with concrete physical artefact's behavior in a playful manner. Moreover, seeing the similarity of the basic micro:bit code structure, in particular, "on start" and "forever" loops (see Figure 10), this is quite handy when introducing the basic components of the Arduino code, which is crucial in terms of assistance during the transfer between virtual and physical realms.

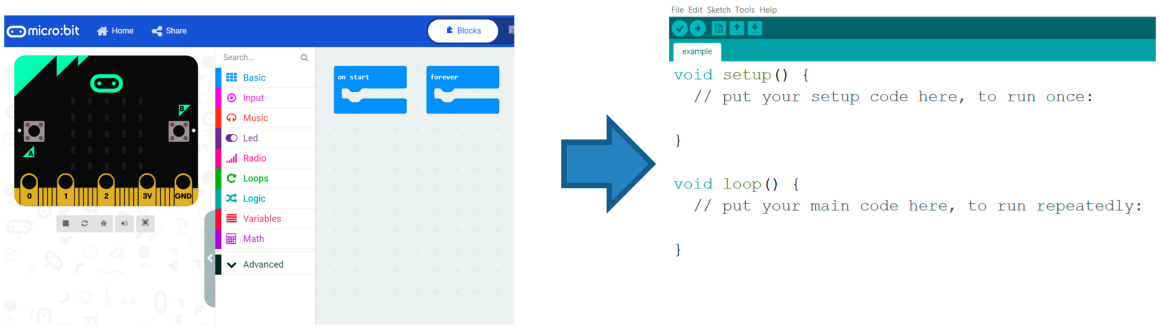


Figure 10. The comparison of micro:bit and Arduino basic code elements representation ("On start" and "Forever" of mircro:bit turn into "void setup ()" and "void loop ()" of Arduino IDE respectively.

In addition, the micro:bit platform was used to enhance the previously acquired computational thinking skills. In this regard, the available guidelines provided by micro:bit educational foundation [56] were partially adopted.

3.3. Data Collection Tools

Given the exploratory character of present research aimed at evaluating the preliminary feedbacks of the target audience to the proposed learning path, the two structured questionnaires on the base of the Coding4Girls approach and tools were administered only for the first section before and after the training course. They were constructed in the framework of a wider validation strategy foreseen in the Coding4Girls project activities according to the following dimensions:

- the programming level evolution;
- the motivation for coding;
- the programming environment usability.

Specifically, the aim was to verify if the proposed pedagogical framework meets the target groups' needs in terms of relevance, acceptance, usability, and effectiveness. Also, the items related to the specificity of the second part, regarding the exploration of educational robotics through the Arduino environment, were integrated into the inquiry.

The qualitative data were collected through observation methods and short interviews. The observation method was led through a grid of open questions. In particular, three aspects were considered in the work:

- the ease of understanding of the concepts to be studied for the target group;
- the achievement of the predefined learning goals for each assignment and the corresponding level of performance;
- the effectiveness of the combination of both methods (CT and ER) and equipment used during the whole designed training path.

The observation was accompanied by a short interview of the students to deepen the three aspects above mentioned and, in particular, when unexpected factors could take place during the accomplishment of the given tasks. For example, in the case of the mental fatigue, if it was caused by prolonged time in school (usually the school time in Italy is from 8:30 a.m. to 13.30 p.m., however, some variations can be found, like from 8:30 a.m. to 4:00 p.m.) or by the difficulty of the tasks at hand.

The analysis of the data gathered through the above-described tools is presented below in Section 5.1.

4. Results

4.1. COVID-19 Emergency Impact

To alleviate the impact of the forced break and to keep contact with the audience, three video lessons were prepared: electric current [57]; robot and its parts [58]; and the control of motors [59]. Videos followed the methodology outlined in the background activities section, i.e., providing the virtual and real representation of the issues considered. The tutorials, as the practice has demonstrated, were quite useful for the underlying information refreshment and the preparation for the final stage of activities. The interested reader is welcomed to get acquainted with the videos by following the links provided [57–59].

Another important effect of the COVID-19 emergency was the organization of the course delivery. After the lockdown, the group was divided into two subgroups of 5 children. The classes were held three times a week in the morning due to the end of the school curricula activities by that moment. It is worth noting that this organization has essentially improved the quality of interaction with the audience.

4.2. Robot Assembly and Programming

The last part of the activities was intended for the purposeful construction of the robot. The work performed by the students was subdivided into the following stages as summarized in Table 1 and Figure 9 below.

Table 1. Overview of the stages for the robotics part.

Stage	Concepts Covered/Activities Performed	Examples of Coding Projects Executed During the Stage
1	A brief introduction to Arduino IDE; The concept of the procedure; The basic components of the Arduino code (void setup(); void loop() supported with the micro:bit environment.	LED blinking
2	Assembling the robot chassis; motor movement inversion through the power supply polarity altering. Visualizing the polarities alternation through the LEDs connected to a couple of Arduino digital outputs. Underlining the difference between the power necessary to control the digital inputs of L298N and the power supplied to move the motors through the corresponding measurements demonstration.	Two LEDs alternating blinking, connected to a couple of Arduino outputs (conceptual demonstration of the motors polarity changing)
3	Working on different combinations of the controlling outputs necessary to move the robot forward, backward, left and right. Adjusting eventual discrepancies between the software procedures (corresponding to left, right, forward, and backward) and the actual physical movement.	Forming the corresponding procedures: left(); right(); forward(); backward(), stop().
4	Introducing the IR remote control, a concept of a library, and remote controller instructions decoding procedure.	Working with the IR remote controller, assignment of the HEX codes to the controlling buttons.
5	Addition of the home-brought auxiliary IR remote controllers to the control of the robotic car; introduction of the OR logical operator.	Putting all together. Control of the car with a couple of different remote controllers.

As one can note from Table 1, the complexity of the activities held during the phase gradually increased. However, despite the seeming difficulty, the passage from a single LED blinking to the development of the code capable of moving a robot-car in four directions is not that long and, therefore, is quite immediate. Figuratively, the related hardware in use, i.e., the L298N bridge, does permit one to concentrate their attention just on the appropriate combination of the "on" and "off" LEDs [59]. As the results have demonstrated (see a top-right image of Figure 11), the children succeeded with the task quite speedily and began to play races, enclosing the forward procedure in a loop.

Certainly, the most challenging passages did require special assistance, particularly during the phases of more advanced coding of IR remote control of the robot. However, the intervention basically consisted of providing examples of the related code at the beginning. Thus, the level of independence of some of the participants in terms of adding the second auxiliary IR remote controller was manifested in their willingness to help their peers (bottom left image of Figure 11).

Figure 11. The final stages of the course: (**top left**)—chassis assembling; (**top right**)—IR control-free basic movements procedures testing; (**bottom left**)—OR operator inclusion—auxiliary IR control involvement; (**bottom right**)—final testing.

5. Discussion

5.1. The Results of Data Analysis

The corresponding results obtained are summarized in Table 2 and the diagrams below. Regarding the programming level (Table 2), it is worth mentioning that nobody selected the option "I was/am able to design a solution of a problem in the form of a program" neither before nor after the course. Despite the last being quite indicative in terms of self-positioning, consciousness, and sincerity, the general trend demonstrated, as one can see, a positive evolution of the skills development (comparing before (blue) and after (green) options checked) in each particular case.

Table 2. The programming level evolution of respondents.

No. of a Student	I Have Never Used Coding or Programming Before	I Was/Am a Beginner Programmer (Have Basic Ideas)	I Could/Can Code Simple Programs	I Was/Am Able to Program (Create a Complete Program)
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

As to the motivation for coding, the options offered were the following:

(a). I am not motivated;

(b). I want to succeed in classroom programming;
(c). I want to demonstrate to other students that I can code;
(d). I want to pursue a career in programming;
(e). I enjoy solving logic problems and puzzles.

The results showed that only one respondent declare having no motivation, while the most popular option selected revealed the major propensity (6 respondents) towards solving logic problems and puzzles. Ultimately, the desire to demonstrate other students the capacity of coding was revealed as suitable for one person only, while the idea of pursuing a career in programming was shown to be attractive to three of the respondents.

Regarding the learning methodology and the usability of the programming environment the diagrams below contain the options provided and the corresponding number of respondents. Note, the options that acquired the positive responses only ("Strongly agree"—SA, "Agree"—Agree), namely: "I found the programming motivating" (4—SA, 6—A), "I have enjoyed programming" (8—SA, 2—A), "Learning in this way is fun" (9—SA, 1—A)—of the learning methodology, and "The tools were easy to use" (3—SA, 7—A)—of the usability of the programming environment) are not included in the general diagram.

As can be seen (Figure 12), just two respondents perceived programming as a challenging task, while the majority, taking into account the bar chart related to the easiness of programming perceived coding, due to the methodology followed, as feasible. As a consequence, the programming concepts covered were proven to understandable. Besides, whereas the presented results of the two last dimensions (relevance and the clearness) provide enough space to perform future work, the commitment to the way of learning, in turn, may in principle, justify the appropriateness of the used methodological and didactical tools.

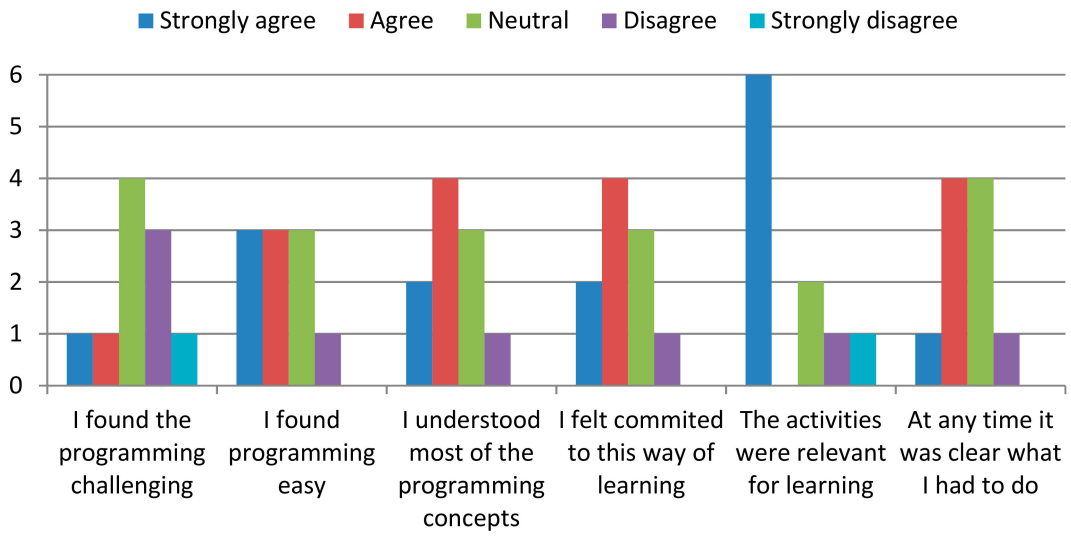


Figure 12. Attitude towards programming.

The first bar chart of the usability of the programming environment diagram (Figure 13) confirms the previous conclusions, i.e., the majority of the respondents expressed a desire to participate in these activities frequently. The symmetry of the next bar chart (the complexity of the activities), undoubtedly reflects the relative difficulty of the tasks, especially during the last stage, and may be attributed to the whole set of the activities performed, and in particular to the programming environment (Arduino IDE) itself (see the corresponding bar chart), which certainly is relatively difficult to get used to at this age. It is worth underlining that this first-hand experience was aimed at rather the conceptual introduction into the matter rather than forcing the audience to master all the intricacies of Arduino programming, which is impossible at this age.

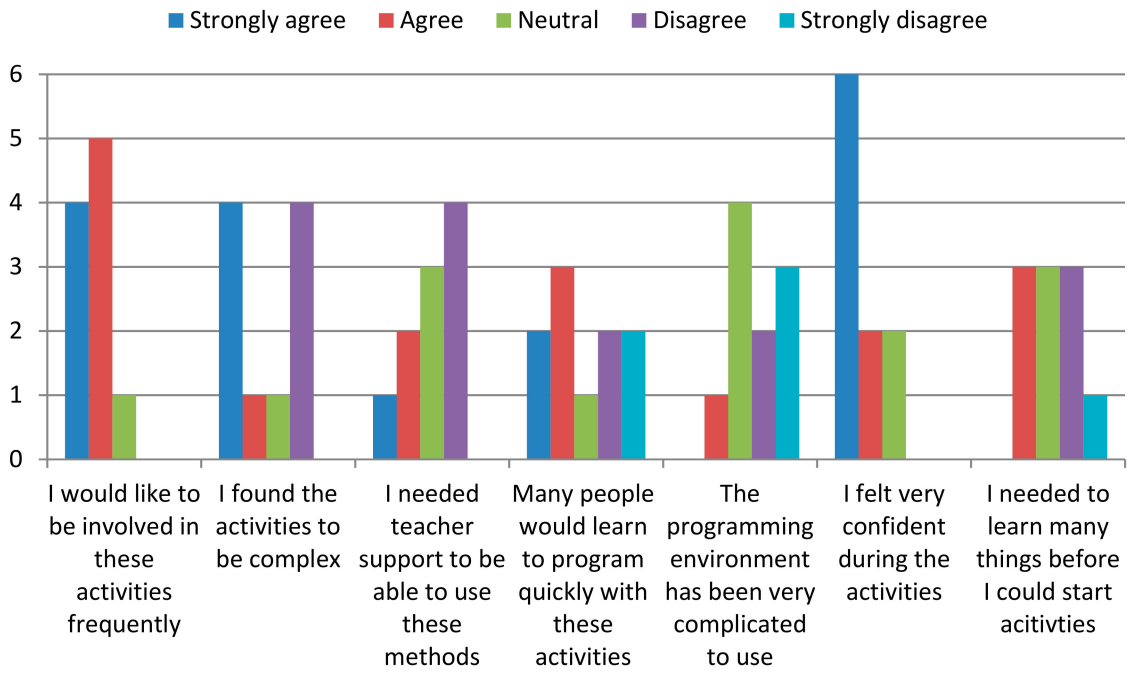


Figure 13. Usability of the programming environment.

Ultimately, this shows that all the beginning work, meant as preliminary work on coding skills development through the Coding4Girls methodology and tools, contributed to preparing students to perform more complex tasks with Arduino IDE.

Therefore, the general approach seems to be amply justified, especially taking into account the trends of the bar charts representing the need for teacher support and confidence during the activities. Surprisingly, the need to learn many things before starting the activities was an issue for three participants only. Whereas the most controversial issue, regarding the feasibility of the course in terms of coding skills acquisition rapidity, could have been caused by the multiperspectiveness of the activities, where coding is seen in conjunction with the hardware and, as a consequence, as a part of a broader spectrum of the concepts covered.

In response to three basic questions outlined in the Research Design section and according to the results achieved, the authors conclude:

- the activities held were perceived quite positively, given the majority had expressed the desire to be involved in such kind of events and the level of the confidence revealed during the first phase (Figure 13); as well as the characteristic tenacity in achieving the set goal (an IR-controlled robot car assembly and programming, Figure 11);
- the typical features of the most attractive instruments are the following: immediate feedback (micro:bit [55], PhET interactive simulations [52]); entertaining/gaming context ("Crack the circuit" [51], building the resistor networks in groups (Figure 9b)); and visual interactivity (PhET interactive simulations [52]).

Regarding the approach to assist the transfer between virtual and physical realms, the authors found appropriate the gradual familiarization with basic programming concepts through Snap! activities followed by the subsequent introduction of the physical concepts and artefacts. Once the basics of coding are grasped and children have witnessed numerous cause-effect manifestations in different programming scenarios contexts, the next step is to gradually shift their attention towards the physical realm. This can be achieved by combining both in vivo demonstrations and using the appropriate virtual counterparts (using, e.g., [52]), allowing children to try them independently while having fun in a safe environment. In this way the physical phenomena and artefacts are seen and manipulated from different perspectives, thus significantly enhancing the underlying concept

acquisition. Finally, to converge virtual and physical realms, the gradual introduction of microntroller programming, starting from user-friendly micro:bit and ending with first steps in Arduino IDE, seems to be didactically appropriate and justified given the results achieved.

6. Conclusions

The course outcomes have demonstrated the feasibility of the adopted methodology for the students to reach the final goal while staying motivated for programming and entertained at the same time. Moreover, the tools involved have been characterized as easy to use, mainly for the Students' Game Environment. On the base of the first results achieved, the proposed methodology favors the development of students' potential because it provides immediate and concrete applications. This contributes to the competences and knowledge construction in STEM from one side and in entrepreneurship and language from the other.

Nevertheless, the final survey has underlined that some small weakness has to be taken into account in the future implementation of the methodology. For example, the long-forced pause due to COVID-19 emergency interrupted the learning flow which would have gained by the face-to-face modality and the personal interaction with the teacher and peers. Of course, the personal preferences of each student were the dominant factor influencing the results in this "intermittent" experience.

Undoubtedly, the gained experience and the current quality of the projects accomplished may furnish the solid fundament for future implications in the project's development and the didactic path. Being modular and highly adjustable, the training structure, constituted of both the coding preparatory section (on the base of the Coding4Girls platforms) and robot's development, permits a wide range of further advancements throughout the whole learning path. While the platform flexibility enables the creation of a truly unlimited study path with different complexity levels by responding the students' real needs, the second component of the course, the creation and development of the robotic elements, with little effort, may see further advancement through the introduction of additional sensors set, functionality, and behavior scenarios. In view of the responses received to the initially posed exploratory questions, the experience gained will certainly play a fundamental role for the next editions of the course.

Thus, the future work will be dedicated to the further enhancement of the concepts covered with particular attention towards the interrelation between the algorithms complexity, microcontroller programming, and the corresponding response of the hardware, as well as by involving a major number of young users in the testing phase to have more complete feedback on the proposed approach. Thanks to the results achieved and the increased and sustained motivation and interest of students at the end of the training, the course will be proposed again with a larger number of participants, but still subdivided into the groups of 5 to follow the general lines of Ministry of Education due to epidemic issues, from November 2020 onwards.

Author Contributions: Conceptualization, M.T. and A.D.; methodology, M.T.; writing—original draft preparation, M.T. and A.D.; writing—review and editing, M.T.; supervision, D.A.; project administration, M.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was co-funded by the European Commission, grant number 2018-1-SI01-KA201-047013.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wing, J. Computational thinking's influence on research and education for all. *Ital. J. Educ. Technol.* **2017**, *25*, 7–14.
2. Bocconi, S.; Chioccariello, A.; Dettori, G.; Ferrari, A.; Engelhardt, K. *Developing Computational Thinking in Compulsory Education-Implications for Policy and Practice*; Joint Research Centre: Seville Site, Spain, 2016.
3. Papert, S. *Mindstorms: Children, Computers, and Powerful Ideas*; Basic Books Inc.: New York, NY, USA, 1980.
4. Papert, S.; Harel, I. Situating constructionism. *Constructionism* **1991**, *36*, 1–11.

5. Wing, J. Research notebook: Computational thinking—What and why? The link. *Mag. Carnegie Mellon Univ. Sch. Comput. Sci.* **2011**, *1*, 2019.
6. Wing, J.M. Computational thinking and thinking about computing. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* **2008**, *366*, 3717–3725. [CrossRef] [PubMed]
7. Shute, V.J.; Sun, C.; Asbell-Clarke, J. Demystifying computational thinking. *Educ. Res. Rev.* **2017**, *22*, 142–158. [CrossRef]
8. Rowe, P.G. *Design Thinking*; MIT Press: Cambridge, MA, USA, 1987.
9. Razzouk, R.; Shute, V. What is design thinking and why is it important? *Rev. Educ. Res.* **2012**, *82*, 330–348. [CrossRef]
10. Dorst, K. The core of 'design thinking' and its application. *Des. Stud.* **2011**, *32*, 521–532. [CrossRef]
11. Rauth, I.; Köppen, E.; Jobst, B.; Meinel, C. Design thinking: An educational model towards creative confidence. In Proceedings of the DS 66-2: 1st International Conference on Design Creativity (ICDC 2010), Kobe, Japan, 29 November–1 December 2010.
12. Brown, T. Design thinking. *Harv. Bus. Rev.* **2008**, *86*, 84.
13. Dunne, D.; Martin, R. Design thinking and how it will change management education: An interview and discussion. *Acad. Manag. Learn. Educ.* **2006**, *5*, 512–523. [CrossRef]
14. Lindberg, T.; Meinel, C.; Wagner, R. Design thinking: A fruitful concept for IT development? In *Design Thinking. Understanding Innovation*; Meinel, C., Leifer, L., Plattner, H., Eds.; Springer: Berlin/Heidelberg, Germany, 2011. [CrossRef]
15. Luka, I. Design thinking in pedagogy. *J. Educ. Cult. Soc.* **2014**, *5*, 63–74. [CrossRef]
16. Dolak, F.; Uebernickel, F.; Brenner, W. Design thinking and design science research. *Commun. Comput. Inf. Sci.* **2013**, *388*, 38–48.
17. Jun, S.; Han, S.; Kim, S. Effect of design-based learning on improving computational thinking. *Behav. Inf. Technol.* **2017**, *36*, 43–53. [CrossRef]
18. Saritepeci, M. Developing Computational Thinking Skills of High School Students: Design-Based Learning Activities and Programming Tasks. *Asia Pac. Educ. Res.* **2020**, *29*, 35–54. [CrossRef]
19. Heidmann, O.; Katsimentes, S.; Panagiotopoulos, H.; Tsalapatas, H. Programming for girls: A game-based approach that deploys design thinking principles. In Proceedings of the Edulearn20: 12th International Conference on Education and New Learning Technologies, Palma de Mallorca, Spain, 6–7 July 2020. [CrossRef]
20. Rich, P.J.; Browning, S.F.; Perkins, M.K.; Shoop, T.; Yoshikawa, E.; Belikov, O.M. Coding in K-8: International Trends in Teaching Elementary/Primary Computing. *TechTrends* **2019**, *63*, 311–329. [CrossRef]
21. Wagner, S.P. Robotics and children: Science achievement and problem solving. *J. Comput. Child. Educ.* **1998**, *9*, 149–192.
22. Barak, M.; Zadok, Y. Robotics projects and learning concepts in science, technology and problem solving. *Int. J. Technol. Des. Educ.* **2009**, *19*, 289–307. [CrossRef]
23. Norton, S.J.; McRobbie, C.J.; Ginns, I.S. Problem solving in a middle school robotics design classroom. *Res. Sci. Educ.* **2007**, *37*, 261–277. [CrossRef]
24. Kamga, R.; Romero, M.; Komis, V.; Mirsili, A. Design requirements for educational robotics activities for sustaining collaborative problem solving. In Proceedings of the International Conference EduRobotics 2016, Athens, Greece, 25 November 2016; pp. 225–228.
25. Skorinko, J.L.; Doyle, J.K.; Tryggvason, G. Do goals matter in engineering education? An exploration of how goals influence outcomes for FIRST robotics participants. *J. Pre-Coll. Eng. Educ. Res.* **2012**, *2*, 3. [CrossRef]
26. Plaza, P.; Sancristobal, E.; Carro, G.; Garcia-Loro, F.; Blazquez, M.; Castro, M. European Robotics Week to introduce robotics and promote engineering. *Comput. Appl. Eng. Educ.* **2018**, *26*, 1068–1080. [CrossRef]
27. Phan, M.-H.; Ngo, H.Q.T. A Multidisciplinary Mechatronics Program: From Project-Based Learning to a Community-Based Approach on an Open Platform. *Electronics* **2020**, *9*, 954. [CrossRef]
28. García-Valcárcel-Muñoz-Repiso, A.; Caballero-González, Y.-A. Robotics to develop computational thinking in early Childhood Education. *Commun. Media Educ. Res. J.* **2019**, *27*, 63–72. [CrossRef]
29. Atmatzidou, S.; Demetriadis, S. Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robot. Auton. Syst.* **2016**, *75*, 661–670. [CrossRef]
30. Chalmers, C. Robotics and computational thinking in primary school. *Int. J. Child Comput. Interact.* **2018**, *17*, 93–100. [CrossRef]

31. Leonard, J.; Buss, A.; Gamboa, R.; Mitchell, M.; Fashola, O.S.; Hubert, T.; Almughyirah, S. Using robotics and game design to enhance children's self-efficacy, STEM attitudes, and computational thinking skills. *J. Sci. Educ. Technol.* **2016**, *25*, 860–876. [CrossRef]
32. Constantinou, V.; Ioannou, A. Development of Computational Thinking Skills through Educational Robotics. In Proceedings of the EC-TEL Practitioner 2019: 14th European Conference on Technology Enhanced Learning, Delft, The Netherlands, 16–19 September 2019.
33. Eguchi, A. Computational thinking with educational robotics. In Proceedings of the Society for Information Technology & Teacher Education International Conference, Savannah, GA, USA, 21 March 2016; pp. 79–84.
34. Vlahu-Gjorgievska, E.; Videnovik, M.; Trajkovik, V. Computational Thinking and Coding Subject in Primary Schools: Methodological Approach Based on Alternative Cooperative and Individual Learning Cycles. In Proceedings of the 2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE), Wollongong, Australia, 4–7 December 2018; pp. 77–83.
35. Atmatzidou, S.; Demetriadis, S. How to support students' computational thinking skills in educational robotics activities. In Proceedings of the 4th International Workshop Teaching Robotics, Teaching with Robotics & 5th International Conference Robotics in Education, Padova, Italy, 18 July 2014; pp. 43–50.
36. Menges, A.; Ahlquist, S. *Computational Design Thinking: Computation Design Thinking*; John Wiley & Sons: Hoboken, NJ, USA, 2011.
37. Repenning, A.; Webb, D.; Ioannidou, A. 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, Milwaukee, WI, USA, 10–13 March 2010; pp. 265–269.
38. Penmetcha, M.R. Exploring the Effectiveness of Robotics as a Vehicle for Computational Thinking. Master's Thesis, Purdue University, West Lafayette, IN, USA, 2012.
39. García-Peñalvo, F.J.; Conde, M.Á.; Gonçalves, J.; Lima, J. Computational thinking and robotics in education. In Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality, Salamanca, Spain, 24–26 October 2018; pp. 2–5.
40. Tramonti, M.; Dochshanov, A. Students' engagement through computational thinking and robotics. In Proceedings of the Digital Presentation and Preservation of Cultural and Scientific Heritage, Burgas, Bulgaria, 24–26 September 2020; pp. 213–219.
41. Valls, A.; Albó-Canals, J.; Canaleta, X. Creativity and contextualization activities in educational robotics to improve engineering and computational thinking. In Proceedings of the International Conference on Robotics and Education RiE 2017, Sofia, Bulgaria, 26–28 April 2017; pp. 100–112.
42. Leonard, J.; Barnes-Johnson, J.; Mitchell, M.; Unertl, A.; Stubbe, C.R.; Ingraham, L. Developing Teachers' Computational Thinking Beliefs and Engineering Practices through Game Design and Robotics. In Proceedings of the 39th Annual Meeting of the North American Chapter of the International Group for the Psychology of Mathematics Education, Indianapolis, IN, USA, 5–8 October 2017; pp. 1289–1296.
43. Cross, J.; Hamner, E.; Zito, L.; Nourbakhsh, I. Engineering and computational thinking talent in middle school students: A framework for defining and recognizing student affinities. In Proceedings of the 2016 IEEE Frontiers in Education Conference (FIE), Eire, PA, USA, 12–15 October 2016; pp. 1–9.
44. Pérez-Marín, D.; Hijón-Neira, R.; Bacelo, A.; Pizarro, C. Can computational thinking be improved by using a methodology based on metaphors and scratch to teach computer programming to children? *Comput. Hum. Behav.* **2020**, *105*, 105849. [CrossRef]
45. Tsarava, K.; Leifheit, L.; Ninaus, M.; Román-González, M.; Butz, M.V.; Golle, J.; Trautwein, U.; Moeller, K. Cognitive correlates of computational thinking: Evaluation of a blended unplugged/Plugged-in course. In Proceedings of the 14th Workshop in Primary and Secondary Computing Education, Glasgow, Scotland, 23–25 October 2019.
46. Ehsan, H.; Rehmat, A.P.; Cardella, M.E. Computational thinking embedded in engineering design: Capturing computational thinking of children in an informal engineering design activity. *Int. J. Technol. Des. Educ.* **2020**. [CrossRef]
47. Bers, M.U.; Flannery, L.; Kazakoff, E.R.; Sullivan, A. Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Comput. Educ.* **2014**, *72*, 145–157. [CrossRef]
48. Coding4Girls Project's Site, Results Section. Available online: https://www.coding4girls.eu/results_02.php (accessed on 16 July 2020).

49. Alimisis, D.; Loukatos, D.; Zoulias, E.; Alimisi, R. The Role of Education for the Social Uptake of Robotics: The Case of the eCraft2Learn Project. In Proceedings of the International Conference on Inclusive Robotics for a Better Society, Pisa, Italy, 16–20 October 2018; pp. 180–187.
50. Dochshanov, A.M.; Lapina, M. Robotics in STEM education: A multiperspective strategy case study. In Proceedings of the 2nd Workshop on Innovative Approaches in Computer Science within Higher Education, Ekaterinburg, Russia, 25–26 November 2019.
51. "Crack the Circuit" Online Game. Available online: https://universeandmore.com/crackthecircuit/ (accessed on 16 July 2020).
52. PhET Interactive Simulations. Available online: https://phet.colorado.edu/ (accessed on 16 July 2020).
53. Chladni Plate Experiment Demonstration. Available online: https://www.youtube.com/watch?v=eb8E9EXLOhU (accessed on 16 July 2020).
54. Function Generator App. Available online: https://play.google.com/store/apps/details?id=com.keuwl.functiongenerator&hl=en (accessed on 16 July 2020).
55. Micro: Bit Educational Foundation's Web-Site. Available online: https://microbit.org/projects/make-it-code-it/ (accessed on 16 July 2020).
56. Zapata-Ros, M. Computational thinking unplugged. *Educ. Knowl. Soc.* **2019**, *20*, 1–29. [CrossRef]
57. Electric Current Tutorial. Available online: https://www.youtube.com/watch?v=FAgqEqRUro4&t=30s (accessed on 2 September 2020).
58. The Robot and Its Parts. Available online: https://www.youtube.com/watch?v=d62EYo0To7Q&t=14s (accessed on 2 September 2020).
59. The Control of Motors. Available online: https://www.youtube.com/watch?v=JlsRXmlhL9I&t=33s (accessed on 2 September 2020).