Multimodal Technologies and Interaction

MDPI

Article

Unplugged Activities in Cross-Curricular Teaching: Effect on Sixth Graders' Computational Thinking and Learning Outcomes

José Miguel Merino-Armero, José Antonio González-Calero *, Ramón Cózar-Gutiérrez and Javier del Olmo-Muñoz

Faculty of Education of Albacete, University of Castilla-La Mancha (UCLM), 02071 Albacete, Spain; josemiguel.merino@alu.uclm.es (J.M.M.-A.); ramon.cozar@uclm.es (R.C.-G.); javier.olmo@uclm.es (J.d.O.-M.)
* Correspondence: jose.gonzalezcalero@uclm.es

Abstract: There is a debate about the way to introduce computational thinking (CT) in schools. Different proposals are on the table; these include the creation of new computational areas for developing CT, the introduction of CT in STEM areas, and the cross-curricular integration of CT in schools. There is also concern that no student should be left behind, independently of their economic situation. To this effect, an unplugged approach is the most cost-effective solution. In addition, this topic is interesting in the context of a pandemic situation that has prevented the sharing of materials between students. This study analyzes an unplugged cross-curricular introduction of CT in the Social Sciences area among sixth grade students. A group of 14 students was selected to carry out an unplugged intervention design—where they were required to program an imaginary robot on paper—in the Social Sciences area. Their CT development and academic results were compared to those of 31 students from the control group who continued attending regular classes. Results showed that an unplugged teaching style of CT in Social Sciences lessons significantly increased CT ($p < 0.001$) and with a large effect size ($d = 1.305$) without differences in students' academic achievement. The findings show that children can potentially develop their CT in non-STEM lessons, learning the same curricular contents, and maintaining their academic results.

Keywords: unplugged; K–6; education; cross-curricular; robotics; programming; computational thinking

check for updates

Citation: Merino-Armero, J.M.; González-Calero, J.A.; Cózar-Gutiérrez, R.; del Olmo-Muñoz, J. Unplugged Activities in Cross-Curricular Teaching: Effect on Sixth Graders' Computational Thinking and Learning Outcomes. *Multimodal Technol. Interact.* **2022**, *6*, 13. https://doi.org/10.3390/mti6020013

Academic Editor: Mu-Chun Su

Received: 22 October 2021
Accepted: 26 January 2022
Published: 28 January 2022

1. Introduction

Computational thinking (CT) is a concept of growing importance [1]. It is booming because technology is the present and the future of our societies [2], and its use is continuously increasing [3]. Digitalization is one of the current priorities of different social agents and states. After the COVID-19 crisis, states are investing in digitalization in pursuit of their economic recovery, especially in Europe with their Next Generation EU funds. Nevertheless, digitalization faces some important challenges, especially in terms of security, data protection, and functionality. In addition, digitalization of work usually requires robots, artificial intelligence, software, and devices to create, control, and understand them. In this context, societies will need citizens with the necessary skills to understand, create, and control technology, and the development of CT will help to face these demands [4].

CT is useful to increase our understanding of technology and it is considered essential for K–12 (kindergarten to grade 12) students [5,6]. As a result, it is gaining relevance in education all over the world [7]. Computer science and CT are being studied at the educational policy level [3,8–12], and teachers at all educational levels have been encouraged to work together with researchers to study and promote CT teaching in different stages of the education process and through different ways (see e.g., [13–16]). These proposals offer very different approaches to the learning of CT in K–12 education. For example, while some of them integrate literacy coding and CT as cross-disciplinary elements in curricula (see [13]), others recommend a specific subject (or area) to promote coding languages (see [16]). In this respect, Swaid [3] points out that some areas of knowledge are more appropriate to develop students' CT because they require its use, such as STEM (science, technology, engineering, and mathematics) subjects.

There are benefits and drawbacks to each of these proposals, but all of them seem effective in their interventions [16]. Brackmann et al. [17] pointed out that there are two main approaches to CT teaching that are being adopted and investigated in schools: one with computer programming exercises (plugged activities) and another with unplugged activities. The difference between them is the way children are exposed to CT; the latter being without using devices [18]. As not all countries nor social classes have the same purchasing power, this situation presents an issue to equality between countries and social strata based on financial variables. There is also concern that no student should be left behind, regardless of their economic situation [14,19]. In this regard, an unplugged approach seems to be the best choice since it is the most cost-effective as no gadget is necessary throughout the teaching–learning process.

In addition, the recent pandemic situation posed a serious challenge in educational settings as students were not allowed to share materials due to health security concerns. Bearing in mind the lack of CT interventions in non-STEM subjects [20,21], and in order to give response to the demands of further evidence regarding the optimal introduction age of CT [14] and the unplugged CT approaches [22,23], this paper aims to analyze the impact of an unplugged approach to CT in a cross-curricular design on CT development and academic results in the area of Social Sciences.

1.1. Background

1.1.1. Computational Thinking

CT is a contemporary concept reformulated from ideas from the 1960s [7]. Alan Perlis reflected on the importance for all university students, regardless of their degrees and areas of study, to be able to program and understand the "theory of computing" [24]. In 1967, Papert and his group created LOGO, a programming language for schoolchildren with the aim of promoting their "procedural thinking" [25], a precursor of CT. However, these ideas were developed perhaps too soon for society's embracement of technology at the time. Technology's applications were reserved for complex calculations. This specificity in its use and the lack of resources available to the general public for accessing the existing technology reduced interest in this type of project. Nevertheless, technology is now widely employed around the world, and a new concept, CT, was brought to the fore by Wing [26], a term that was defined as a process that "involves problem-solving, system design and understanding of human behavior" [26] (p. 33).

After some considerations from the scientific community, Wing [27] redefined CT as "the thought processes involved in formulating a problem and expressing its solution(s) in such a way that a computer (human or machine) can effectively carry it out" (p. 8). A similar definition to that was specified by INTEF [28] in Spain, defining CT as "the ability to solve problems and communicate ideas taking advantage of the power offered by computers" (p. 16).

Programming, according to Grover and Pea [29], is a key tool for supporting cognitive tasks connected to computational thinking. Therefore, code-literacy skills are understood to be an important part of STEM disciplines [30]. One of the most relevant curricular definitions of CT comes from the MIT-Harvard model [15], which is mainly oriented to the use of their own block-based programming language, Scratch. However, CT and programming should not be considered to be the same thing [15,31]. CT is a way of thinking that can be used in most of the tasks people usually carry out [32,33]. For instance, Henderson [34] claims that "writing instructions, choreographing a dance, using graphical software, cooking using a recipe, following instructions to construct a table or using an electronic instrument are all examples of everyday computational thinking" (p. 100).

In this study, according to INTEF [28], the MIT-Harvard model was selected to design the experimental interventions devoted to teaching CT. According to this model, CT is divided into three computational dimensions: concepts, practices, and perspectives. Computational concepts are sequences, loops, events, parallelism, conditionals, operators, and data. For example, when cooking: one must sequence the order of the ingredients; to avoid food burning, one must stir it every minute or so (loop); if the dish tastes bland, one must add salt (conditional), etc. Computational practices are those processes that routinely ensure success during programming. They are composed of experimentation and iteration; evaluation and debugging; reuse and remixing; and abstraction and modularization. The last dimension refers to those changes of perspective that students appreciate from learning CT, these being: to express oneself, to connect, and to question. Expressing yourself, because programming is used as a means of creation, design, and self-expression; connecting, so that people share their work and connect with the community, developing collaborative works; and questioning, that is asking about the digital world closest to them.

1.1.2. Computational Thinking through Unplugged Activities

Unplugged teaching of CT and computer science (CS) is not new (see [35]). However, studies about this type of teaching have rapidly increased in recent years [36]. Most of these papers compare plugged and unplugged activities in CT learning [5,6,37]. Bell and Vahrenhold [38] discern that the approaches of unplugged activities relate to CT in education in the following ways: (1) supporting or complementing computer programming—the most popular way to develop CT—; (2) helping with integrating CT with other subjects, such as math, physics, biology, or music; (3) teachers learning about CT for professional development; and (4) measuring CT achievement in the form of Bebras challenges.

It should be noted that plugged and unplugged approaches to CT are not mutually exclusive, as several studies have shown (see [39–43]). Both approaches can achieve different goals or be complementary and reinforcing (see e.g., [44–48]). In particular, an idea that is gaining ground, and supported by solid evidence from research, is that unplugged activities are an appropriate prior step to understanding algorithmic thinking and procedures before using programming languages [49–54]. What is more, it seems that unplugged teaching of CT is more effective than plugged teaching at early ages and when CT is initially introduced [17,19,55]. There are initiatives based on unplugged teaching of CT, such as CSUnplugged.org, a program developed by the CS Education Research Group of Canterbury University in New Zeeland, which provides guidance and materials for teachers and scholars for students' development of their CS skills. CSUnplugged.org consists of a website translated to multiple languages with collections of activities. Another noteworthy initiative is Code.org, whose aim is to expand the access to CS in schools through a website that includes a combination of coding lessons and unplugged activities that has proven to successfully promote CT (see [55]).

1.2. Objectives

The aim of this study is twofold: first, to evaluate whether unplugged activities linked to Social Sciences curricular contents can be effective in promoting elementary school students' CT skills; and second, to analyze the effect of this approach on the teaching and learning of these contents. The experimental intervention was intentionally designed based on the following principles: (i) do not modify the time devoted to the teaching of the Social Sciences concepts, and (ii) adapt the introduction of CT to the teaching of these contents by making as few changes as possible to their teaching without CT. Under this approach, our hypothesis is that the intervention should allow students to improve their level of CT and, at the same time, their academic performance should not be affected. The area of Social Sciences was selected with the aim of generating evidence about the viability of making the development of CT compatible with the teaching of non-STEM disciplines. Thus, the study analyses the effect of an unplugged CT instruction integrated with the contents of the Social Sciences subject. In particular, after the intervention, we evaluated the results in terms of students' development of CT and academic achievement of students in that subject. The research objectives are as follows:

- Objective 1: To analyze eventual differences in sixth graders' development of CT skills after an unplugged CT intervention integrated in the Social Sciences area, compared to regular Social Sciences lessons.
- Objective 2: To analyze eventual differences in the sixth graders' Social Sciences academic achievement after an unplugged CT intervention integrated in the Social Sciences area, compared to regular Social Sciences lessons.

2. Materials and Method

This study employed a posttest-only control group design [56], and, accordingly, followed a four-step methodology: (1) selection and distribution of participants into two conditions (experimental and control); (2) administration of intervention; (3) gathering data for assessment; and (4) comparison of results between groups to determine the effectiveness of the intervention.

2.1. Participants

Students from three sixth-grade classes at a Spanish school participated in the study. Given that the experiment was developed in a pandemic context, and in accordance with strict school regulations aimed at preventing the spread of the virus and avoiding new confinements, under no circumstances could pupils from different groups be mixed. Thus, one of the groups was randomly assigned to the experimental condition, while the other two groups formed the control condition. The experimental and control conditions consisted of 14 and 31 students, respectively.

In Spain, one of the measures that the educational administration implemented to return to face-to-face lessons after lockdown was to reduce the ratio of students per classroom. In the school where the study was conducted, before the pandemic, there were only two sixth-grade classes, but this measure implied that these two groups had to be reorganized to form three groups of fewer students. In this process, the school management team tried to make the three groups comparable in terms of academic achievement, gender, and number of students. Accordingly, the selection of the participants after this distribution process helped to minimize sampling bias [57].

2.2. Intervention

The interventions in the experimental and control conditions were similar insofar as both had the same length and followed the same curricular program (teaching–learning standards, contents, etc.). However, in the experimental condition, some of the tasks were replaced by activities related to CT development integrated with the knowledge area. These activities were unplugged and versed over programming imaginary robots. These differences were mainly related to practical activities. The intervention lasted 6 weeks and consisted of two teaching units: "The orographic relief of Spain" and "The orographic relief of Europe". Both teaching units addressed contents related to mountains, rivers, and coasts of different geographical regions. Each week, students had three lessons of 45 min each. In the experimental condition, there were neither extra time nor sessions for additional explanations.

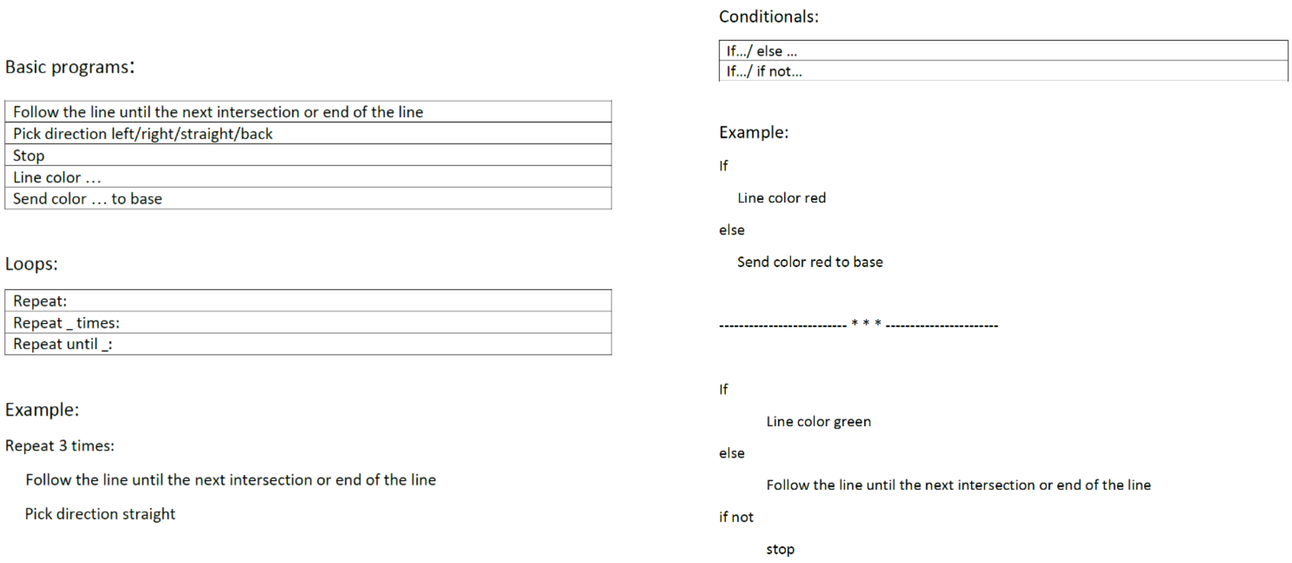
The activities in the experimental condition revolved around how to program an imaginary spatial robot (Figure 1) that could only move on a grid of meridians and parallels of the Earth. This robot had to carry out the recognition of the territory for supporting special agents in their missions over Spanish and European geography. Students were the programmers of this imaginary robot and were instructed to direct it from point A to B by stopping over diverse geographical accidents as mountain ranges, rivers, and coasts of Spain and Europe using the orders that they preliminary had learned.

We need you!

There are strategic places where machines are our best allies. The space is one of these locations. But robots do what people program them to do. So, they need excellent programmers who know how to place the instructions correctly to complete special missions. Our special agents are on the ground and need support. Your time has come. Below you have the necessary instructions to program the space robot. Good luck!

Figure 1. The introduction to the imaginary spatial robot activity.

They also had a guide with the basic instructions for programming the robot (Figure 2). The activities were inspired by the type of movements and programming language of the educational robot Ozobot.



Basic programs:

Follow the line until the next intersection or end of the line
Pick direction left/right/straight/back
Stop
Line color …
Send color … to base

Loops:

Repeat:
Repeat _ times:
Repeat until _:

Example:

```
Repeat 3 times:
    Follow the line until the next intersection or end of the line
    Pick direction straight
```

Conditionals:

If…/ else …
If…/ if not…

Example:

```
If
    Line color red
else
    Send color red to base

------------------------- * * * ----------------------

If
        Line color green
else
        Follow the line until the next intersection or end of the line
if not
        stop
```

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

Figure 2. Instructions for students to program the spatial robot: (**a**) basic programs and loops, with an example; and (**b**) conditionals with examples.

An example of an activity outline is "Our special agents have missions on Sierra Morena, Sistema Central, and the Pirineos, and to this effect, they need the support of the spatial robot. Program it for completing the recognition of the territory to accomplish our mission. The spatial robot is located at coordinates (6, 2). Look at the map (Figure 3) and, good luck!"

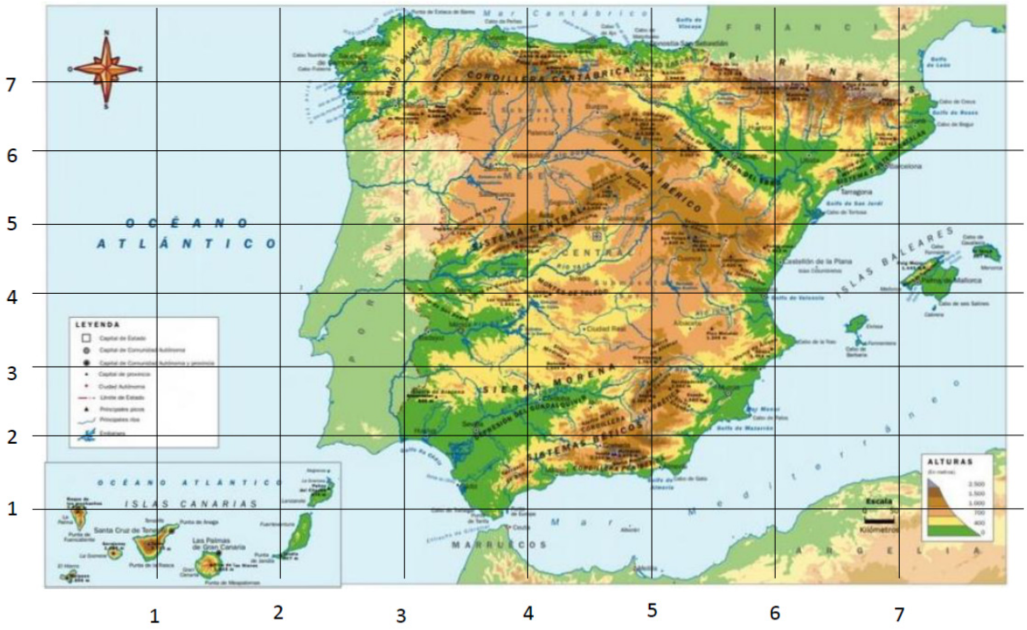


Figure 3. Grid map used in one of the activities.

The activities were distributed throughout the first two units on geography in the Social Sciences area, during the months of September, October, and November.

2.3. Assessment

To evaluate the effect of the intervention, two variables were chosen: students' CT skills and academic achievement in Social Sciences. For that purpose, we used two instruments: CT was assessed with the computational thinking test (CTt) [58], and the academic achievement in the area of Social Sciences was determined by the students' grades in the evaluation tests employed by the school to assess students' content knowledge (the same for both conditions). CTt was completed at the end of the intervention, and the academic achievement instrument was made up of two tests, one for "The orographic relief of Spain" and another for "The orographic relief of Europe". The first academic achievement test was completed in the third week and the second one was completed in the sixth week (Table 1).

Table 1. Time sequence of methodological design.

Week 1 to Week 3		Week 4 to Week 6		Week 7
Teaching Unit 1 (TU1): The orographic relief of Spain	Academic achievement test of TU1	Teaching Unit 2 (TU2): The orographic relief of Europe	Academic achievement test of TU2	CTt

The CTt was distributed in a color-printed paper format, validated to evaluate this type of thinking in sixth-grade students [58,59]. The instrument is made up of 28 items formulated from images based on a journey through a maze, with 4 multiple-choice options to answer the questions, with the appearance of visual block programming where only one of the answers is correct.

The tests for measuring the academic results were two tests made up of 10 questions each. These assessments included curricular contents from the two corresponding teaching units in Social Sciences. These tests did not have questions about programming or CT.

2.4. Analysis

All the tests were anonymized before coding with the purpose of avoiding interviewer bias [57]. The questions were scored binarily, that is, if the answer was correct, 1 point was assigned; if it was incorrect, 0 points were given. Finally, the data were analyzed with the SPSS 24 software. First, the normality and homoscedasticity of each sample distribution were analyzed; then, two *t*-tests were carried out, one for the academic results and another for CT results. In addition, effect sizes of both variables were calculated. In particular, Cohen's *d* was employed as a measure of the effect size.

3. Results

First, due to the reduced number of participants of the study, Shapiro–Wilk normality tests were used to assess the assumption of normality in each condition for each outcome variable (CT and academic scores). The test revealed that both groups fulfilled this assumption ($p > 0.05$) for both variables. Then, Levene tests were conducted to study the homogeneity of variances in the outcome variables, and both confirmed the assumption of homoscedasticity ($p > 0.05$). To compare academic results, the average score of the two academic achievement tests was used (Table 2). A first glance at the results shows higher results in both variables for the intervention group, with greater differences in CT than in the case of academic scores.

Table 2. CT and academic achievement average scores.

Group	N	CT Scores	Academic Scores
Intervention	14	15.14 (3.28)	6.43 (2.39)
Control	31	10.94 (3.19)	5.97 (2.43)

To establish whether these differences were significant, two Student's t-tests were performed. Results of these tests revealed that the differences in CT scores were statistically significant ($t(43) = -4.059$; $p < 0.001$), but not in academic scores ($t(43) = -0.589$; $p = 0.559$). This means that teaching Social Sciences with unplugged robotics activities increased students' CT skills. It also means that the unplugged instruction did not affect students' achievement, either positively or negatively, compared to the control instruction.

Finally, effect sizes were calculated. Concerning CT scores, the differences in favor of the experimental condition can be considered large-sized, according to Cohen [60], as $d = 1.31$. Regarding the differences in academic performance, the effect size was small ($d = 0.19$) according to Cohen, but medium-sized corresponding to Kraft's scale [61]. According to the last scale measure, which establishes different grades of effect size taking into account the relation cost–benefit of treatments, this intervention has a medium effect size because the economic cost of the intervention for pupils is in the lower range and gains are near to a value of $d = 0.2$. However, taking into account Cohen's standards, although the results are slightly higher for the experimental group, the effect size should be considered as small.

4. Discussion

The analysis of results shows that teaching CT with an unplugged cross-curricular approach has been found to be effective for the development of CT skills. The effect size is large ($d = 1.31$), according to Cohen [60]. In addition, according to Kraft's scale [61], where the relation cost–benefit is the primary factor, the effect of intervention on the academic achievement is medium-sized. These results are in concordance with previous studies [17,19,55], whose analyses provided evidence of the high influence and results of unplugged pedagogy in CT development at elementary levels. Therefore, the argument for unplugged CT education as the foundation for understanding before creating codes [49–54] is gaining force and these studies could support it.

Furthermore, unplugged programming activities can enhance educational equity in CT education, since their cost-effectiveness enables the full integration of students from different backgrounds in the teaching–learning process. This is especially important for countries and regions where the lack of hardware facilities, the economy, or their geographical barriers do not allow equal access to this knowledge, as Sun et al. [19] highlighted. In addition, something remarkable is the fact that, in a pandemic context, it was possible to teach CT with an unplugged approach without sharing materials and without peer contact. Even in the most difficult situations, education must continue, and the unplugged methodology has been postulated as an interesting approach for teaching CT.

Regarding academic achievement, no statistically significant differences were observed between experimental and control conditions, although scores were slightly higher (small effect size) for those students who attended unplugged CT teaching. In this respect, it should be underlined that Social Sciences is a subject that does not belong to the STEM disciplines, and that, therefore, has less connection with CT, according to Swaid [3]. Despite the apparent relationship between CT and STEM subjects, the role of CT in non-STEM subjects is multifaceted and promising, even though it is still underexplored [62]. In this context, our results should be seen as highly positive, as they illustrate how children can effectively develop their CT in non-STEM lessons, learning the same curricular contents as in lessons in which the promotion of CT is not involved. Moreover, the present study is a contribution that will help to fill the gap in research addressing unplugged approaches, as reported by Kite et al. [22], who reviewed the trends and issues in CT education research, or by Huang and Looi [23], who reviewed the literature on unplugged pedagogies in K–12 CS and CT education.

Limitations

This study has some limitations, such as the reduced number of participants. The sample size should be increased in future studies. Additionally, it is necessary to carry out more studies about the effects of unplugged pedagogy on the development of CT and on the learning of curricular contents in different areas and grades. If these contents were reduced and/or academic results worsened, it would be necessary to consider whether learning CT could be at the expense of another kind of knowledge. This piece of research has only studied one of the curricular areas and only within a limited period of time, so the information is reduced and focused only on Social Sciences. Studies in other areas or of longer duration will be necessary to completely understand the scope of this methodology.

5. Conclusions

In this study, unplugged teaching of CT has been demonstrated to be completely effective with a large effect size, without harming students' academic achievement. The relevance of these findings is enhanced by the fact that material costs are minimal, formative courses for teachers being one of the few requirements of its large-scale implementation. Therefore, all countries and districts should at least be able to afford minimal and initial CT development. In addition, this approach did not affect academic results, either positively or negatively. In this respect, it must be considered that Social Sciences are not included in STEM areas and that benefits on these areas could be achieved from CT instruction, according to other studies [63,64]. With one eye on the demands of society and, taking into account the vast volume of evidence to which this study contributes, it seems that there are no excuses for CT instruction to be implemented in every primary school. Equity in CT education is difficult to achieve, but cross-curricular unplugged teaching can ease the way.

Author Contributions: Conceptualization, J.M.M.-A. and R.C.-G.; methodology, J.M.M.-A., J.A.G.-C., R.C.-G. and J.d.O.-M.; software, J.M.M.-A., J.A.G.-C., R.C.-G. and J.d.O.-M.; validation, J.M.M.-A., J.A.G.-C., R.C.-G. and J.d.O.-M.; formal analysis, J.M.M.-A. and J.A.G.-C.; investigation, J.M.M.-A. and R.C.-G.; resources, J.M.M.-A., J.A.G.-C., R.C.-G. and J.d.O.-M.; data curation, J.M.M.-A., J.A.G.-C., R.C.-G. and J.d.O.-M.; writing—original draft preparation, J.M.M.-A., J.A.G.-C., R.C.-G. and J.d.O.-M.; writing—review and editing, J.M.M.-A., J.A.G.-C., R.C.-G. and J.d.O.-M.; visualization, J.M.M.-A.,

J.A.G.-C., R.C.-G. and J.d.O.-M.; supervision, J.M.M.-A.; project administration, J.A.G.-C. and R.C.-G.; funding acquisition, J.A.G.-C., R.C.-G. and J.d.O.-M. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the University of Castilla-La Mancha under grant 2021-GRIN-31060; by the European Union through European Regional Development Funds (ERDF-FEDER) and the Castilla-La Mancha Regional Administration under grant SBPLY/19/180501/000278; and by the Ministry of Education, Culture and Sports of Spain under grant FPU19/03857.

Institutional Review Board Statement: The study was conducted according to the guidelines of the Declaration of Helsinki. Ethical review and approval were waived for this study because the project was a cooperation of researchers, teachers, students, external experts, and the school management. All participants agreed to participate before the project started.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Data available upon reasonable request.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Shute, V.J.; Sun, C.; Asbell-Clarke, J. Demystifying computational thinking. *Educ. Res. Rev.* **2017**, *22*, 142–158. [CrossRef]
2. Iversen, O.S.; Smith, R.C.; Dindler, C. From Computational Thinking to Computational Empowerment. In Proceedings of the 15th Participatory Design Conference, Hasselt and Genk, Belgium, 20–24 August 2018; ACM: New York, NY, USA, 2018; Volume 1, pp. 1–11.
3. Swaid, S.I. Bringing Computational Thinking to STEM Education. *Procedia Manuf.* **2015**, *3*, 3657–3662. [CrossRef]
4. Mishra, P.; Yadav, A.; Henriksen, D.; Kereluik, K.; Terry, L.; Fahnoe, C.; Terry, C. Rethinking Technology & Creativity in the 21st Century. *TechTrends* **2013**, *57*, 10–14. [CrossRef]
5. Lye, S.Y.; Koh, J.H.L. Review on teaching and learning of computational thinking through programming: What is next for K-12? *Comput. Hum. Behav.* **2014**, *41*, 51–61. [CrossRef]
6. Buitrago Flórez, F.; Casallas, R.; Hernández, M.; Reyes, A.; Restrepo, S.; Danies, G. Changing a Generation's Way of Thinking: Teaching Computational Thinking Through Programming. *Rev. Educ. Res.* **2017**, *87*, 834–860. [CrossRef]
7. Angeli, C.; Valanides, N. Developing young children's computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy. *Comput. Hum. Behav.* **2020**, *105*, 105954. [CrossRef]
8. INTEF. *Programación, Robótica y Pensamiento Computacional en el Aula*; INTEF: Madrid, Spain, 2018.
9. Manches, A.; Plowman, L. Computing education in children's early years: A call for debate. *Br. J. Educ. Technol.* **2017**, *48*, 191–201. [CrossRef]
10. Micheuz, P. Some findings on informatics education in Austrian academic secondary schools. *Inform. Educ.* **2008**, *7*, 221–236. [CrossRef]
11. The Royal Society. *Shut Down or Restart? The Way Forward for Computing in UK Schools*; The Royal Academy of Engineering: London, UK, 2012.
12. Angeli, C.; Voogt, J.; Fluck, A.; Webb, M.; Cox, M.; Malyn-Smith, J.; Zagami, J. A K-6 computational Thinking curriculum framework: Implications for teacher knowledge. *J. Educ. Technol. Soc.* **2016**, *19*, 47–57. [CrossRef]
13. Balanskat, A.; Engelhardt, K. *Computing Our Future. Computer Programming and Coding. Priorities, School Curricula and Initiatives across Europe*; European Schoolnet: Brussels, Belgium, 2015.
14. Bocconi, S.; Chioccariello, A.; Dettori, G.; Ferrari, A.; Engelhardt, K.; Kampylis, P.; Punie, Y. *Developing Computational Thinking in Compulsory Education—Implications for Policy and Practice*; European Commision: Brussels, Belgium, 2016.
15. Brennan, K.; Resnick, M. New Frameworks for Studying and Assessing the Development of Computational Thinking. In Proceedings of the Annual American Educational Research Association Meeting, Vancouver, BC, Canada, 13–17 April 2012; pp. 1–25.
16. Merino-Armero, J.M.; González-Calero, J.A.; Cózar-Gutiérrez, R. Computational thinking in K-12 education. An insight through meta-analysis. *J. Res. Technol. Educ.* **2021**, 1–26. [CrossRef]
17. Brackmann, C.P.; Román-González, M.; Robles, G.; Moreno-León, J.; Casali, A.; Barone, D. Development of Computational Thinking Skills through Unplugged Activities in Primary School. In Proceedings of the 12th Workshop on Primary and Secondary Computing Education—WiPSCE'17, Nijmegen, The Netherlands, 8–10 November 2017; ACM Press: New York, NY, USA, 2017; pp. 65–72.
18. Bell, T.; Alexander, J.; Freeman, I.; Grimley, M. Computer Science Unplugged: School Students Doing Real Computing Without Computers. *J. Appl. Comput. Inf. Technol.* **2009**, *13*, 20–29.

19. Sun, L.; Hu, L.; Zhou, D. Improving 7th-graders' computational thinking skills through unplugged programming activities: A study on the influence of multiple factors. *Think. Ski. Creat.* **2021**, *42*, 100926. [CrossRef]
20. Cutumisu, M.; Adams, C.; Lu, C. A Scoping Review of Empirical Research on Recent Computational Thinking Assessments. *J. Sci. Educ. Technol.* **2019**, *28*, 651–676. [CrossRef]
21. Tang, X.; Yin, Y.; Lin, Q.; Hadad, R.; Zhai, X. Assessing computational thinking: A systematic review of empirical studies. *Comput. Educ.* **2020**, *148*, 103798. [CrossRef]
22. Kite, V.; Park, S.; Wiebe, E. The Code-Centric Nature of Computational Thinking Education: A Review of Trends and Issues in Computational Thinking Education Research. *SAGE Open* **2021**, *11*, 215824402110164. [CrossRef]
23. Huang, F.; Teo, T. Examining the role of technology-related policy and constructivist teaching belief on English teachers' technology acceptance: A study in Chinese universities. *Br. J. Educ. Technol.* **2021**, *52*, 441–460. [CrossRef]
24. Guzdial, M. Education: Paving the way for computational thinking. *Commun. ACM* **2008**, *51*, 25. [CrossRef]
25. Papert, S. *Mindstorms: Children, Computers and Powerful Ideas*; Basic Books: New York, NY, USA, 1983; Volume 1, ISBN 0465046274.
26. Wing, J.M. Computational thinking. *Commun. ACM* **2006**, *49*, 33. [CrossRef]
27. Wing, J.M. Computational thinking's influence on research and education for all Influenza del pensiero computazionale nella ricerca e nell'educazione per tutti. *Ital. J. Educ. Technol.* **2017**, *25*, 7–14. [CrossRef]
28. Ministerio de Educación y Formación Profesional. *La Escuela de Pensamiento Computacional y su Impacto*; INTEF: Madrid, Spain, 2019.
29. Grover, S.; Pea, R. Computational Thinking in K–12. *Educ. Res.* **2013**, *42*, 38–43. [CrossRef]
30. Weintrop, D.; Beheshti, E.; Horn, M.; Orton, K.; Jona, K.; Trouille, L.; Wilensky, U. Defining Computational Thinking for Mathematics and Science Classrooms. *J. Sci. Educ. Technol.* **2016**, *25*, 127–147. [CrossRef]
31. Moreno, J.; Robles, G.; Román, M.; Rodríguez, J.D. No es lo mismo: Un análisis de red de texto sobre definiciones de pensamiento computacional para estudiar su relación con la programación informática. *Rev. Interuniv. Investig. Tecnol. Educ.* **2019**, *7*, 1–10. [CrossRef]
32. Balladares, J.A.; Avilés, M.R.; Pérez, H.O. Del pensamiento complejo al pensamiento computacional: Retos para la educación contemporánea. *Sophia Colecc. Filos. Educ.* **2016**, *1*, 143–159. [CrossRef]
33. Valverde-Berrocoso, J.; Fernández-Sánchez, M.R.; Garrido-Arroyo, M.C. El pensamiento computacional y las nuevas ecologías del aprendizaje. *Rev. Educ. Distancia* **2015**, *9*, 1–18. [CrossRef]
34. Henderson, P.B. Ubiquitous Computational Thinking. *Computer* **2009**, *42*, 100–102. [CrossRef]
35. Bell, T.; Witten, I.H.; Fellows, M. Computer Science Unplugged. Off-Line Activities and Games for All Ages. Available online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.114.7908&rep=rep1&type=pdf (accessed on 26 January 2022).
36. Huang, W.; Looi, C.-K. A critical review of literature on "unplugged" pedagogies in K-12 computer science and computational thinking education. *Comput. Sci. Educ.* **2021**, *31*, 83–111. [CrossRef]
37. Ribeiro Silva, L.; da Silva, A.P.; Toda, A.; Isotani, S. Impact of Teaching Approaches to Computational Thinking on High School Students: A Systematic Mapping. In Proceedings of the 18th IEEE International Conference on Advanced Learning Technologies, ICALT 2018, Bombay, India, 9–13 July 2018; pp. 285–289. [CrossRef]
38. Bell, T.; Vahrenhold, J. CS Unplugged—How Is It Used, and Does It Work? In *Adventures between Lower Bounds and Higher Altitudes*; Lecture Notes in Computer Science; Böckenhauer, H.-J., Komm, D., Unger, W., Eds.; Springer: Cham, Switzerland, 2018; Volume 11011, pp. 497–521.
39. Alamer, R.A.; Al-Doweesh, W.A.; Al-Khalifa, H.S.; Al-Razgan, M.S. Programming Unplugged: Bridging CS Unplugged Activities Gap for Learning Key Programming Concepts. In Proceedings of the Fifth International Conference on e-Learning (econf), Manama, Bahrain, 18–20 October 2015; pp. 97–103.
40. Dorling, M.; White, D. Scratch: A Way to Logo and Python. In Proceedings of the 46th ACM Technical Symposium on Computer Science Education, Kansas City, MO, USA, 4–7 March 2015; pp. 191–196. [CrossRef]
41. Faber, H.; Wierdsma, M.; Doornbos, R.P.; Van der Ven, J.S.; de Vette, K. Teaching computational thinking to primary school students via unplugged programming lessons. *J. Eur. Teach. Educ. Netw.* **2017**, *12*, 13–24.
42. Gaio, A. Programming for 3rd Graders, Scratch-Based or Unplugged? In Proceedings of the CERME 10, Dublin, Ireland, 1–5 February 2017.
43. Hermans, F.; Aivaloglou, E. To Scratch or Not to Scratch? A Controlled Experiment Comparing Plugged First and Unplugged First Programming Lessons. In Proceedings of the 12th Workshop on Primary and Secondary Computing Education, Nijmegen, The Netherlands, 8–10 November 2017; ACM: New York, NY, USA, 2017; pp. 49–56.
44. Aggarwal, A.; Gardner-McCune, C.; Touretzky, D.S. Evaluating the Effect of Using Physical Manipulatives to Foster Computational Thinking in Elementary School. In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education, Seattle, WA, USA, 8–11 March 2017; ACM: New York, NY, USA; pp. 9–14.
45. Connor, R.; Cutts, Q.; Robertson, J. Keeping the machinery in computing education. *Commun. ACM* **2017**, *60*, 26–28. [CrossRef]
46. Gunion, K.; Milford, T.; Stege, U. The Paradigm Recursion: Is It More Accessible When Introduced in Middle School? *J. Probl. Solving* **2009**, *2*, 8. [CrossRef]
47. Lonati, V.; Malchiodi, D.; Monga, M.; Morpurgo, A. Is Coding the Way to Go? In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Cham, Switzerland, 2015; Volume 9378, pp. 165–174.

48. Schofield, E.; Erlinger, M.; Dodds, Z. MyCS: CS for Middle-Years Students and Their Teachers. In Proceedings of the 45th ACM Technical Symposium on Computer Science Education, Atlanta, Georgia, USA, 5–8 March 2014; pp. 337–342. [CrossRef]
49. Caeli, E.N.; Yadav, A. Unplugged Approaches to Computational Thinking: A Historical Perspective. *TechTrends* **2020**, *64*, 29–36. [CrossRef]
50. Montes-León, H.; Hijón- Neira, R.; Pérez-Marín, D.; Montes-León, S.R. Improving Computational Thinking in Secondary Students with Unplugged Tasks. *Educ. Knowl. Soc.* **2020**, *21*, 12. [CrossRef]
51. Sun, L.; Hu, L.; Zhou, D. Which way of design programming activities is more effective to promote K-12 students' computational thinking skills? A meta-analysis. *J. Comput. Assist. Learn.* **2021**, *37*, 1048–1062. [CrossRef]
52. Wohl, B.; Porter, B.; Clinch, S. Teaching Computer Science to 5–7 Year-Olds. In Proceedings of the Workshop in Primary and Secondary Computing Education, Nijmegen, The Netherlands, 8–10 November 2017; ACM: New York, NY, USA, 2017; pp. 55–60.
53. Gardeli, A.; Vosinakis, S. Creating the computer player: An engaging and collaborative approach to introduce computational thinking by combining 'unplugged' activities with visual programming. *Ital. J. Educ. Technol.* **2017**, *25*, 36–50. [CrossRef]
54. Uchida, Y.; Matsuno, S.; Ito, T.; Sakamoto, M. A proposal for teaching programming through the Five-Step Method. *J. Robot. Netw. Artif. Life* **2015**, *2*, 153–156. [CrossRef]
55. del Olmo-Muñoz, J.; Cózar-Gutiérrez, R.; González-Calero, J.A. Computational thinking through unplugged activities in early years of Primary Education. *Comput. Educ.* **2020**, *150*, 103832. [CrossRef]
56. Frey, B.B. *The SAGE Encyclopedia of Educational Research, Measurement, and Evaluation*; SAGE Publications: Thousand Oaks, CA, USA, 2018.
57. Bryman, A. *Social Research Methods*; Oxford University Press: Oxford, UK, 2012; ISBN 978-0-19-958805-3.
58. Román-González, M. Códigoalfabetización y Pensamiento Computacional en Educación Primaria y Secundaria: Validación de un Instrumento y Evaluación de Programas, Programa de Doctorado en Educación. Ph.D. Thesis, Universidad Nacional de Educación a Distancia, Madrid, Spain, 2016.
59. Román-González, M.; Pérez-González, J.-C.; Jiménez-Fernández, C. Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Comput. Hum. Behav.* **2017**, *72*, 678–691. [CrossRef]
60. Cohen, J. *Statistical Power Analysis for the Behavioral Sciences*, 2nd ed.; Lawrence Erlbaum Associates: New York, NY, USA, 1988.
61. Kraft, M.A. Interpreting Effect Sizes of Education Interventions. *Educ. Res.* **2020**, *49*, 241–253. [CrossRef]
62. Grover, S.; Pea, R. Computational Thinking: A Competency Whose Time Has Come. In *Computer Science Education*; Bloomsbury Academic: New York, NY, USA, 2018.
63. Sáez-López, J.-M.; Román-González, M.; Vázquez-Cano, E. Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools. *Comput. Educ.* **2016**, *97*, 129–141. [CrossRef]
64. Jenkins, C. Poem Generator: A comparative quantitative evaluation of a microworlds-based learning approach for teaching English. *Int. J. Educ. Dev. Using Inf. Commun. Technol.* **2015**, *11*, 153–167.