



Article

Exploring the Effect of Training in Visual Block Programming for Preservice Teachers

José Manuel Sáez-López ¹, Javier del Olmo-Muñoz ^{2,*}, José Antonio González-Calero ²
and Ramón Cózar-Gutiérrez ²

¹ Faculty of Education, National University of Distance Education (UNED), 28040 Madrid, Spain; jmsaezlopez@edu.uned.es

² Faculty of Education of Albacete, University of Castilla-La Mancha (UCLM), 02071 Albacete, Spain; Jose.gonzalezcalero@uclm.es (J.A.G.-C.); Ramon.Coazar@uclm.es (R.C.-G.)

* Correspondence: Jdelolmo1@gmail.com

Received: 5 August 2020; Accepted: 10 September 2020; Published: 12 September 2020



Abstract: This study evaluates the effectiveness of visual block programming-based instruction and its possibilities in the training of future teachers. In particular, the application Scratch, a visual programming environment, was employed to introduce pre-service teachers to programming. The study followed a mixed-method design with a sample of 79 pre-service teachers. A quantitative approach was used to evaluate the gains in the participants' knowledge of computational concepts and attitudes towards Scratch as a pedagogic tool. A qualitative analysis aimed at evaluating the participants' knowledge concerning programming applications, and their perception about possible difficulties in the implementation of programming in educational contexts. Positive results were obtained for programming in the classroom, with significant improvements in innovation, collaboration, active learning, motivation, and fun for the students. After the experiment, the subjects highlighted Scratch as a fundamental block programming tool and the need for teacher training in this field. The need to improve the implementation of visual block programming in Education Degree curricula is supported.

Keywords: higher education; visual block programming; teacher training; educational technology

1. Introduction

Society and the labor market demand mastery of the essential tools to function effectively in the information age. One of these technology-related tools, commonly known as digital competence, is programming [1] which is a means of creating something new in the digital world, solving problems, implementing ideas and using creativity. Programming provides benefits and advantages in different areas of elementary education, regarding motivation, digital competence and skills to create. It is introduced through block-based visual languages [2], and Scratch is one of the most widely used products worldwide [3]. However, despite growing interest in introducing computational thinking and programming into K-12 contexts [4], most teachers find it difficult to use Scratch in the classroom [3].

Technologies already have a significant impact on learning, teaching and creative thinking in university education [5]. However, in pre-service Teacher Education Programs as well as in primary and secondary education, new approaches are needed to integrate technology into education and train digitally competent teachers [6,7]. The European Higher Education Area aims to clarify and resolve this issue by examining how European Higher Education Institutions generate new teaching formulas, aimed at promoting active and dynamic learning, technology use, and student-centered methodologies in response to changing demands [8].

From an international perspective, these practices in schools are related to K-12 curriculum standards, including the Common Core State Standards [9] and CSTA K-12 Computer Science Standards [10], and other important perspectives of digital competence framed in key competences for lifelong learning.

Thereby, these changes can and should be addressed through the development of digital competence—which is substantially more complex and wide than just coding—within initial training of future teachers, something that will benefit their educational practice and turn them into good role models, which in turn will positively impact on the learning outcomes of their future learners, and respond to the demands of future job opportunities [11]. This implies the need to develop training activities that contemplate the most appropriate spaces and strategies for the correct acquisition of teaching skills, including digital competence [12], which allows the solving of problems and situations in digital environments [13].

In the context of K-12 elementary education, computing began to be considered in educational contexts based on the contributions of Seymour Papert at MIT in the 1980s. This author pioneered the idea of children developing procedural thinking through LOGO programming [14]. Subsequently, Ref. [15] has provided theoretical elements of interest focused on computational thinking. Computational thinking is “the solution of problems, the design of systems and the understanding of human behavior, making use of the fundamental concepts of computer science” [15] (p. 33). Computational methods and models allow us to solve problems and design systems that humans cannot do alone; in short, it is about using computational principles to solve a series of problems. It implies logical analysis and data organization, as well as modelling, abstractions, simulations, and the application of possible solutions. These approaches in the world of education help students acquire essential skills to solve complex problems.

Various authors (e.g., [16]) have emphasized that programming is an extension of writing and, as with traditional writing, there are powerful reasons for everyone to learn to program. In primary education, programming can be applied transversally in many subjects (e.g., mathematics, computer science, language, art, social sciences) so that, from an interdisciplinary perspective, students can learn strategies to solve problems, design projects, and communicate ideas [17].

The visual block programming environment of Scratch provides a considerable advantage over other languages when the aim is to introduce students to computing, as it allows users to “write” by dragging and dropping graphic blocks to develop simple programs that, in turn, allow them to create games, interactive stories, or simulations [18]. Scratch is a visual block programming language created by the Lifelong Kindergarten group in the MIT Media Lab. Scratch’s programming language offers more than 100 programming blocks, grouped into eight different categories (movement, appearance, sound, pencil, data, events, control, detection, operators, and variables). This programming environment allows young people to create their own interactive stories, games, and simulations, and then share those creations in an online community with other young programmers from around the world.

Visual programming languages, specifically Scratch, solve many problems related to the difficulties of introducing programming into educational contexts [19]. Scratch allows a simple and attractive introduction to programming and avoids the difficulties posed by classic textual programming languages [20]. In this context, a study providing detailed evidence related to the integration of visual block programming as combined tools in areas framed in the current curriculum is therefore of great interest. Brennan and Resnick [21] pointed out the main activities to be included in the design of learning environments related to constructionism, which are based on the belief that the most effective learning experiences are related to active construction, socially significant elements [14,22], interactions, and elements supporting reflection [14,23].

Several studies have found positive results related to attitudes towards computer science [24–26] and the improvement of skills related to computational concepts [27,28]. Israel, Pearson, Tapia, Wherfel, and Reese [29] argued that research concerning teaching practices indicates that teachers who were initially skeptical about programming implementation, in the end, considered that the

Scratch application was really valuable and accessible; something that has also been argued by other authors [30].

In a study with elementary students, Sáez-López et al. [2] reported statistically significant improvements in the understanding of computational concepts and practices due to the use of Scratch in the classroom, which suggests recommending educational authorities to implement programming in elementary educational environments. Students can approach computational practices and create their own content related to curricular areas, especially in the social sciences and arts, given the characteristics of the visual content these areas present, allowing the creation of colorful, dynamic, and motivating projects from an active perspective. These conclusions are reaffirmed in the study by [31] which demonstrated the acquisition of basic concepts of computer programming in the elementary education setting.

These investigations highlight the advantages related to student motivation, fun, commitment and enthusiasm through this pedagogical approach. Students are generally in favor of this pedagogical design, highlighting its usefulness and the active learning provided. The importance of an educational design that includes the approach of a visual programming language to understand the elements of logic, mathematics, and the creation of content in art and history, brings about significant improvements both in elementary [2] and higher education [32] through the understanding of computational concepts and active approaches.

From our position as trainers of trainers, and from the application of approaches focused on visual block programming, we consider the impact generated by these emerging innovations in the initial training of future teachers, as well as the real presence of these key trends in the university training context. To address this, we specify the following aims:

- O01—To assess the effect of a visual block programming-based intervention on pre-service teachers' mastery of coding.
- O02—To assess the effect of a visual block programming-based intervention on pre-service teachers' opinions about the potential of programming as a pedagogical tool.
- O03—To assess the pre-service teachers' satisfaction and enjoyment based on the visual block programming-based intervention.

2. Materials and Methods

The study followed a pre-experimental design as educational ethics prevented researchers to configure a control group, which is a clear limitation of the experiment and confers an exploratory nature to this piece of research. The intervention was carried out in two classroom-based instruction sessions of 5 h each, in two different weeks. Additionally, 10 h of autonomous student work were added, which was carried out outside the classroom. Figure 1 summarizes the different phases of the intervention.

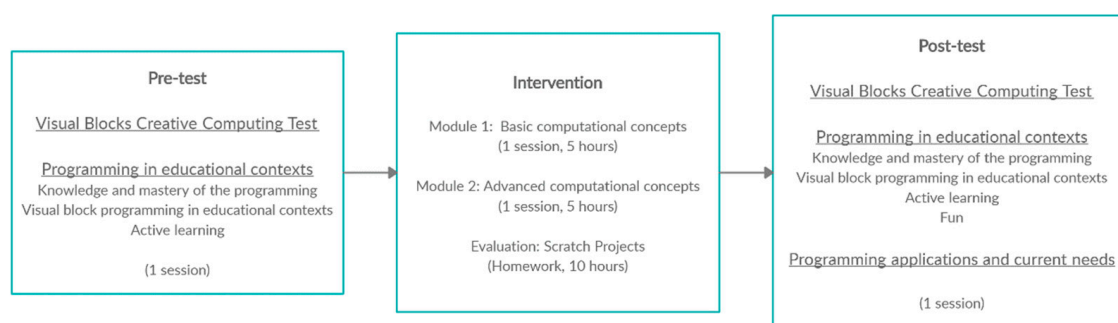


Figure 1. Intervention design.

Before the first session, the students took a pre-test in which their prior knowledge of computational concepts, their opinion on coding in educational contexts and their knowledge of coding applications

and current needs were measured. The instruction was structured in two modules. In the first one, the students were introduced to basic computational concepts (sequences, loops, conditionals) through the development of simple projects using Scratch. In the second module, more advanced computational concepts (parallelism, events, operators and data) were explored, as well as project-design practices through the creation of games where the pedagogical advantages of coding in the teaching of Social Sciences could be highlighted.

The following computational concepts [21] were used in this study:

- Sequence: To create a program it is necessary to think about the order of the steps.
- Iteration (loop): “always” and “repeat” can be used for iteration (repetition of a series of instructions).
- Conditional statements: “if” can be used to establish a condition.
- Threads (parallel execution): Execution of two independent orders that are executed in parallel.
- Event Management: For example, when a key is pressed or when objects are clicked.

Subsequently, the students had to develop, as homework, an original project with *Scratch* in which the concepts and programming learned during the instruction had to be used in order to enhance their creativity. This work would be evaluated by the teacher and would be posted in *Scratch Studio*, where all the projects developed could be reviewed. Once the instruction was finished, the participants were again invited to carry out the post-test following the same procedure used in the pre-test.

2.1. Participants

The study population are university students in the second year of the Degree in Primary Education in the Faculty of Education, Albacete (University of Castilla-La Mancha), enrolled in the subject “Social Sciences II: History and its didactics”. The sample, which was non-probabilistic and intentional, consisted of 79 individuals. The tests from the design are applied as related samples linked to a pre-test/post-test design. The experimental group consisted of 72.2% female students and 27.8% male students, which turns out to be a representative sample of the gender disparity that exists in teacher studies, in which there is always a greater number of women. The average age was almost 21 (20.85). The group was quite homogeneous, because it consisted of students of almost the same age and educational level.

2.2. Instruments

This study uses the survey technique and a questionnaire as instruments, consistent with the objectives of the research. In dimension 1: Results and computational concepts, the Visual Blocks Creative Computing Test (VBCCT) [2] was used. The test was made up of 20 multiple-choice items with 4 options and one correct answer, in which students must answer questions related to computational concepts and computational practices. Figure 2 shows an example of the questions used.

In dimension 2: Programming in educational contexts, a questionnaire with four dimensions adapted from Cózar-Gutiérrez and Sáez-López [33] was used. The questionnaire includes different sub-dimensions: (1) *Knowledge and mastery of the programming*, (2) *Visual block programming in educational contexts*, (3) *Active learning*, and (4) *Fun*. Items in the first two sub-dimensions were adapted from [2], while items for *Active Learning* and *Fun* were adapted from [34,35], respectively. The descriptive data of the four sub-dimensions were assessed from the pre-experimental approach, applying, again, one sample *t*-tests between the pre- and post-test. The results were analyzed through an inferential analysis with one sample *t*-tests, except for sub-dimension 4, *Fun*, which was exclusively measured in the post-test.

14.- Para que aparezca una imagen por unos segundos al recibir la orden, necesitamos: *

- ☐ 1- Mostrar, esconder.
- ☐ 2.- Mostrar, Mostrar.
- ☐ 3.- Esconder, esconder.
- ☐ 4.- Esconder, girar (10) grados.

14



Figure 2. Example of a question from the Visual Blocks Creative Computing Test.

This information allows us to determine if there are significant differences before and after the intervention carried out with the students. The level of significance (α) was 0.05 in all the cases. Reliability calculated through Cronbach's alpha coefficient is 0.83 in the first dimension, and an average of 0.71 for the different scales in the second dimension; as it is always greater than 0.6, it is considered acceptable [36]. In dimension 3: Programming applications and current needs, the qualitative information of open questions is analyzed.

The analysis followed a data triangulation approach [37] using quantitative and qualitative information, which makes it possible to determine whether there is evidence to support the validity of the results and minimize error variance [38]. The data triangulation was implemented using quantitative information collected in the aforementioned test in the first dimension, and the questionnaire in the second dimension. We obtained qualitative data from the open questions in the third dimension.

3. Results

3.1. Dimension 1: Results and Computational Concepts

For dimension 1, a pre-experimental design was applied, analyzing data with the one-sample *t*-test [39]. The Creative Visual Block Computing Test (TVBCC) was administered with a pre-test/post-test design, which makes it possible to check if there were significant improvements after the intervention. The values in the 20-item test emphasize the importance of an educational design that includes a visual programming language to understand the elements of logic, mathematics, and content creation, providing improvements as highlighted in this analysis.

The mean post-test scores for the 79 participants was 11.43 (SD = 2.52), which was compared to the mean pre-test score of 6.59 (SD = 2.18). The results revealed a significant difference between pre- and post-test scores on the TVBCC test ($t = 17.04$, $p < 0.001$, $d = 1.92$). It can thus be affirmed that there were significant gains after the interventions. Indeed, the effect size d is 1.92, which can be considered

as large [40]. Consequently, the implemented program improved the students' ability to understand the concepts and logic of programming to create multimedia products related to the curriculum.

3.2. Dimension 2: Coding in Educational Contexts

To evaluate dimension 2, one-sample *t*-tests were performed for each item (Table 1) and for each sub-dimension. Concerning the first sub-dimension (*Knowledge and mastery of the programming*), the participants' scores were significantly higher in the post-test ($M_{post} = 3.00$, $SD_{post} = 0.47$) compared to the pre-test ($M_{pre} = 2.38$, $SD_{pre} = 0.50$, $t = 11.64$, $p < 0.001$). In addition, the difference can be seen as large ($d = 1.32$). An in-depth analysis of each item in this dimension indicates that pre-service teachers have positive opinions concerning the importance of programming knowledge for their future career, and also in their training at the university (items 1.1, 1.2, and 1.3). The gains in these items are moderate-medium sized (see Table 1). The intervention was particularly effective at increasing the participants' experience in programming (item 1.4), and qualifying them to use block programming languages to create content in educational contexts and use them as classroom tools (items 1.5 and 1.6), with large effect sizes.

Table 1. Descriptive analysis, t-test, and effect size for each item in sub-dimension 2.

Sub-Dimension	Item	M_{pre}	SD_{pre}	M_{post}	SD_{post}	t	p	d
1. Knowledge and mastery of coding	1. Knowledge of programming or coding is essential to train future teachers	3.03	0.88	3.27	0.69	3.03	0.0033	0.35
	2. In initial university training it is important to work on codification	2.94	0.72	3.19	0.62	3.57	<0.001	0.40
	3. The interaction with visual block programming is beneficial in initial teacher training.	3.05	0.82	3.28	0.60	3.40	0.0011	0.38
	4. I have worked with visual block programming in the university.	1.73	0.81	2.68	0.87	9.74	<0.001	1.09
	5. I know how to program in blocks to create content in educational contexts.	1.59	0.73	2.80	0.81	13.30	<0.001	1.49
	6. I know several tools that allow programming in the classroom.	1.92	0.76	2.77	0.78	9.67	<0.001	1.09
2. Visual block programming in educational contexts	1. The visual block programming approach enhances creativity development.	2.85	0.95	3.59	0.63	10.50	<0.001	1.17
	2. Coding work facilitates collaborative advantages.	2.72	0.80	3.32	0.65	8.14	<0.001	0.92
	3. Communication and interactions are improved through visual block programming.	2.62	0.84	3.29	0.64	9.23	<0.001	1.05
	4. The approach of visual block programming enables the development of competencies in educational contexts.	2.75	0.88	3.41	0.61	9.54	<0.001	1.08
	5. The visual block programming approach fosters educational innovation processes.	2.90	0.91	3.59	0.61	10.12	<0.001	1.13
	6. Work with coding increases motivation in learning processes.	2.86	0.83	3.54	0.64	9.56	<0.001	1.06
3. Active learning	1. You learn many contents with visual block programming.	2.72	0.83	3.28	0.64	7.77	<0.001	0.87
	2. The central themes related to the contents are identified through visual block programming.	2.47	0.78	3.01	0.69	7.00	<0.001	0.78
	3. With this coding approach the topic worked is more interesting.	2.82	0.92	3.43	0.65	8.30	<0.001	0.94
	4. With visual block programming you actively participate.	2.81	0.95	3.57	0.67	10.03	<0.001	1.13
4. Fun	1. I was happy.	—	—	3.34	0.64	—	—	—
	2. I really liked the activity.	—	—	3.47	0.64	—	—	—
	3. I was excited.	—	—	3.24	0.66	—	—	—
	4. I felt motivated.	—	—	3.24	0.74	—	—	—
	5. I was relaxed and comfortable.	—	—	3.49	0.62	—	—	—

Concerning the second sub-dimension (*Visual block programming in educational contexts*), the scores revealed that the intervention improves pre-service teachers' opinions in favor of the pedagogic use of these programming languages. Thus, the comparison between pre-test ($M_{pre} = 2.78$, $SD_{pre} = 0.72$) and

post-test scores ($M_{post} = 3.46$, $SD_{post} = 0.46$) showed the huge benefits derived from the intervention in this sub-dimension ($t = 13.07$, $p < 0.001$, $d = 1.48$). After the experiment, the students stressed that visual block programming enhances creativity, collaborative advantages, communication, skills development, innovation and motivation in educational contexts (items 2.1, 2.2, 2.3, 2.4, 2.5, and 2.6). The effect can be seen as large for all these items.

Regarding the third sub-dimension (*Active learning*), the experience led participants to consider that programming activities promote active involvement and learning (items 3.1 and 3.4) and make the subjects more interesting (3.3). Globally, the participants' scores were again significantly higher in the post-test ($M_{post} = 3.32$, $SD_{post} = 0.49$) compared to the pre-test ($M_{pre} = 2.71$, $SD_{pre} = 0.73$, $t = 11.18$, $p < 0.001$, $d = 1.24$).

Finally, the last sub-dimension (*Fun*), which was only measured in the post-test as it exclusively refers to motivational variables related to the activities developed during the experience. A descriptive analysis indicated that the participants felt extremely happy, motivated, relaxed, and comfortable during the activities (see Table 1).

In short, all the inferential tests, together with the descriptive analysis, point to a very positive impact of the intervention in all the sub-dimensions, promoting the students' mastery of coding, their attitude towards visual block programming as an educational tool, and their enjoyment.

3.3. Dimension 3: Programming Applications and Current Needs

A qualitative analysis was conducted based on the participants' answers to a survey. In the design of the instrument, we opted for a mixed questionnaire consisted of two open questions as an instrument because it enables students to respond freely. Considering the purpose of this paper, only the answers of one question were included in the analysis. Indeed, the frequencies provided by the subjects to the question provided elements and factors of interest for the study, which were numbered and accounted for to reinforce the values obtained in the quasi-experimental design and descriptive analysis.

In particular, the question asked students "What problems or obstacles do you perceive to implement visual block programming in educational contexts?" The sample subjects highlighted the need for training, time, and effort in the design and development of these activities, as well as the need for resources and means (Figure 3).

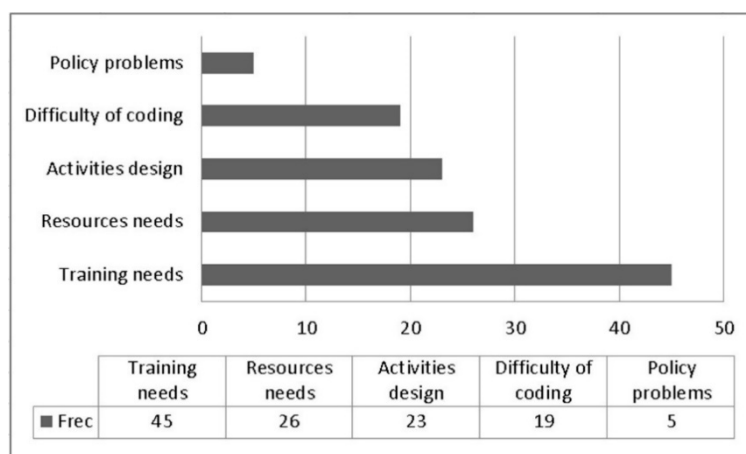


Figure 3. "What problems or obstacles do you perceive for the implementation of visual block programming in educational contexts?".

4. Discussion

Many studies in the last few years have analyzed visual block programming in educational contexts and have yielded positive results similar to those found in our study [2,3,29,31]. Based on triangulation of the results, it can be concluded, in a structured way, that:

1. The subjects positively valued receiving training in programming during their initial teacher training. Based on the TVBCC values, statistically significant improvements were obtained after the program was implemented, improving the ability of university students to understand the concepts and logic of programming to create multimedia products as part of their training to be future teachers (dimension 1).

2. The formative action significantly favored knowledge about, attitudes towards, and the application of programming in initial stages of teacher training. The subjects considered programming to be essential in the training future teachers, and very important during initial university education (items 1.1, 1.2, and 1.3).

3. Although many of the participants had experience in programming before the intervention, pre-service teachers improved their skills in the implementation of visual block programming after the intervention, which points in the same direction as the results obtained in [3].

4. Block programming in educational contexts (sub-dimension 2) showed positive and significant values for the importance of creativity (item 2.1), interactive, collaborative advantages, and the development of competencies (items 2.2, 2.3, and 2.4). Educational innovation processes were also possible (item 2.5), and, as in [2], the motivation of the students was increased and greatly enhanced (item 2.6).

5. With high values and significant improvements, it appears that integrating programming into the classroom encourages active learning (sub-dimension 3), while emphasizing that coding allows the topic being worked on to be more interesting (item 3.3), and programming encourages active participation and learning of the essential lesson content (items 3.1, 3.2, and 3.4).

6. These practices provided the students with satisfaction and fun (sub-dimension 4), and the students stated that they were happy, motivated and liked the programming activities (items 4.1, 4.2, 4.3, 4.4, and 4.5).

7. The main obstacle to implementing block programming appeared to be the need for teacher training, due to the difficulty of these pedagogical approaches and computational practices, something that may be related to the pedagogical beliefs of teachers about the barriers in the use of technology in the classroom [41] and their perceived competence [42]. To a lesser extent, the need for material resources and the difficulty in curricular design for these types of activities were also mentioned, which underpins what was previously denoted in [29].

Once the availability of resources, class planning and initial and continuous teacher training have been overcome, the implementation of visual block programming in educational contexts creates pedagogical advantages that allow greater student activity and prominence, taking advantage of the strength of interest and motivation that these approaches elicit, through collaboration, communication, creativity and student satisfaction.

International studies and reports have proposed that these key trends be adopted in the short term, leading to a change in educational practices [1]. University education and initial teacher training must adapt to these challenges and the demands of today's society, taking into account the emerging trends that prospective professionals will encounter in their immediate future [33]. This and other studies [43,44] provide evidence concerning the attitudes, assessments, and perspectives of the application of programming in university contexts among future teachers. Due to the difficulty of ensuring adequate application, and the essential need for training to avoid the barriers and difficulties that arise due to erroneous pedagogical application, now is a key moment for the introduction of educational technology during initial teacher training in university contexts, since if such programming training is delayed, the opportunity to incorporate these technologies into the classroom will be lost. How to teach *Scratch* at elementary classrooms is beyond the scope of this study as the training was focused on developing a good command in *Scratch*, but it was not aimed at teaching the future teachers how to introduce this visual language to elementary students. However, this issue is essential to computer sciences education and should be addressed in future studies. This study highlights the positive assessment of programming in the classroom by future teachers during their training period,

with special focus on the advantages in terms of motivation, innovation, collaboration, active learning, and student satisfaction. Highlighting Scratch as a fundamental tool, and teacher training in this field as a challenge to overcome, visual block programming in higher education should be a reality that fosters the integral training of future teachers.

Author Contributions: Conceptualization, J.M.S.-L. and R.C.-G.; methodology, J.M.S.-L., J.d.O.-M., J.A.G.-C. and R.C.-G.; software, J.M.S.-L., J.d.O.-M., J.A.G.-C. and R.C.-G.; validation, J.M.S.-L., J.d.O.-M., J.A.G.-C. and R.C.-G.; formal analysis, J.M.S.-L. and J.A.G.-C.; investigation, J.M.S.-L. and R.C.-G.; resources, J.M.S.-L., J.d.O.-M., J.A.G.-C. and R.C.-G.; data curation, J.M.S.-L., J.d.O.-M., J.A.G.-C. and R.C.-G.; writing—original draft preparation, J.M.S.-L., J.d.O.-M., J.A.G.-C. and R.C.-G.; writing—review and editing, J.M.S.-L., J.d.O.-M., J.A.G.-C. and R.C.-G.; visualization, J.M.S.-L., J.d.O.-M., J.A.G.-C. and R.C.-G.; supervision, J.M.S.-L.; project administration, J.A.G.-C. and R.C.-G.; funding acquisition, J.A.G.-C. and R.C.-G. 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 2020-GRIN-28755 and by the Castilla-La Mancha Regional Administration under grant SBPLY/19/180501/000278.

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. Nouri, J.; Zhang, L.; Mannila, L.; Norén, E. Development of computational thinking, digital competence and 21 st century skills when learning programming in K-9. *Educ. Inq.* **2020**, *11*, 1–17. [CrossRef]
2. 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]
3. Marcelino, M.J.; Pessoa, T.; Vieira, C.; Salvador, T.; Mendes, A.J. Learning Computational Thinking and scratch at distance. *Comput. Hum. Behav.* **2018**, *80*, 470–477. [CrossRef]
4. 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]
5. Garrison, R.; Akyol, Z. Role of instructional technology in the transformation of higher education. *J. Comput. High. Educ.* **2009**, *21*, 19–30. [CrossRef]
6. Instefjord, E.J.; Munthe, E. Educating digitally competent teachers: A study of integration of professional digital competence in teacher education. *Teach. Teach. Educ.* **2017**, *67*, 37–45. [CrossRef]
7. Kay, R.H. Evaluating strategies used to incorporate technology into preservice education: A review of the literature. *J. Res. Technol. Educ.* **2006**, *38*, 383–408. [CrossRef]
8. Gaebel, M.; Zhang, T.; Bunesco, L.; Stoeber, H. *Learning and Teaching in the European Higher Education Area*; European University Association: Brussels, Belgium, 2018.
9. National Governors Association. *Common Core State Standards*; National Governors Association: Washington, DC, USA, 2010.
10. Computer Science Teachers Association. *CSTA K-12 Computer Science Standards (Revised 2017)*; Computer Science Teachers Association: New York, NY, USA, 2017.
11. Gudmundsdottir, G.B.; Hatlevik, O.E. Newly qualified teachers’ professional digital competence: Implications for teacher education. *Eur. J. Teach. Educ.* **2018**, *41*, 214–231. [CrossRef]
12. Esteve Mon, F.; Adell Segura, J.; Gisbert Cervera, M. Diseño de un entorno 3D para el desarrollo de la competencia digital docente en estudiantes universitarios: Usabilidad, adecuación y percepción de utilidad. *RELATEC* **2014**, *13*, 35–47.
13. Aviram, A.; Eshet-Alkalai, Y. Towards a Theory of Digital Literacy: Three Scenarios for the Next Steps. *Eur. J. Open Distance E Learn.* **2006**, *9*. Available online: https://www.eurodl.org/materials/contrib/2006/Aharon_Aviram.htm (accessed on 11 September 2020).
14. Papert, S. *Mindstorms: Children, Computers, and Powerful Ideas*; Basic Books: New York, NY, USA, 1980.
15. Wing, J.M. Computational thinking. *Commun. ACM* **2006**, *49*, 33. Available online: <http://www.cs.cmu.edu/~jw/15110-s13/Wing06-ct.pdf> (accessed on 11 September 2020). [CrossRef]
16. Resnick, M.; Maloney, J.; Monroy-Hernández, A.; Rusk, N.; Eastmond, E.; Brennan, K.; Kafai, Y. Scratch: Programming for all. *Commun. ACM* **2009**, *52*, 60–67. [CrossRef]

17. Rodríguez-Martínez, J.A.; González-Calero, J.A.; Sáez-López, J.M. Computational thinking and mathematics using Scratch: An experiment with sixth-grade students. *Interact. Learn. Environ.* **2020**, *28*, 316–327. [CrossRef]
18. Maloney, J.H.; Peppler, K.; Kafai, Y.; Resnick, M.; Rusk, N. Programming by choice. In Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education—SIGCSE '08, Portland, OR, USA, 12–15 March 2008; ACM: New York, NY, USA, 2008; pp. 367–371.
19. Ouahbi, I.; Kaddari, F.; Darhmaoui, H.; Elachqar, A.; Lahmine, S. Learning Basic Programming Concepts by Creating Games with Scratch Programming Environment. *Procedia Soc. Behav. Sci.* **2015**, *191*, 1479–1482. [CrossRef]
20. Kaučič, B.; Asič, T. Improving introductory programming with Scratch? In Proceedings of the 2011 34th International Convention MIPRO, Opatija, Croatia, 23–27 May 2011; pp. 1095–1100.
21. Brennan, K.; Resnick, M. New frameworks for studying and assessing the development of computational thinking. In Proceedings of the 2012 Annual Meeting of the American Educational Research Association, Vancouver, BC, Canada, 13–17 April 2006.
22. Bruckman, A. Learning in Online Communities. In *The Cambridge Handbook of the Learning Sciences*; Sawyer, K., Ed.; Cambridge University Press: New York, NY, USA, 2006; pp. 461–472.
23. Kolodner, J.L.; Camp, P.J.; Crismond, D.; Fasse, B.; Gray, J.; Holbrook, J.; Ryan, M. Problem-Based Learning Meets Case-Based Reasoning in the Middle-School Science Classroom: Putting Learning by Design™ into Practice. *J. Learn. Sci.* **2003**, *12*, 495–547. [CrossRef]
24. Lambert, L.; Guiffre, H. Computer science outreach in an elementary school. *J. Comput. Sci. Coll.* **2009**, *24*, 118–124.
25. Lin, J.M.C.; Yen, L.Y.; Yang, M.C.; Chen, C.F. Teaching Computer Programming in Elementary Schools: A Pilot Study. In Proceedings of the 2005 National Educational Computing Conference, Philadelphia, PA, USA, 20–26 June 2005.
26. Phillips, R.S.; Brooks, B.P. *The Hour of Code: Impact on Attitudes towards and Self-Efficacy with Computer Science*; Code: Seattle, WA, USA, 2017.
27. 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]
28. Kwon, D.Y.; Kim, H.S.; Shim, J.K.; Lee, W.G. Algorithmic Bricks: A tangible robot programming tool for elementary school students. *IEEE Trans. Educ.* **2012**, *55*, 474–479. [CrossRef]
29. Israel, M.; Pearson, J.N.; Tapia, T.; Wherfel, Q.M.; Reese, G. Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Comput. Educ.* **2015**, *82*, 263–279. [CrossRef]
30. Clark, J.; Rogers, M.P.; Spradling, C.; Pais, J. What, no canoes? Lessons learned while hosting a scratch summer camp. *J. Comput. Sci. Coll.* **2013**, *28*, 204–210.
31. Sáez-López, J.M.; Cózar Gutiérrez, R. Programación visual por bloques en Educación Primaria: Aprendiendo y creando contenidos en Ciencias Sociales. *Rev. Complut. Educ.* **2017**, *28*, 409–426. Available online: <http://revistas.ucm.es/index.php/RCED/article/view/49381> (accessed on 11 September 2020). [CrossRef]
32. Cetin, I. Preservice Teachers' Introduction to Computing: Exploring Utilization of Scratch. *J. Educ. Comput. Res.* **2016**, *54*, 997–1021. [CrossRef]
33. Cózar-Gutiérrez, R.; Sáez-López, J.M. Game-based learning and gamification in initial teacher training in the social sciences: An experiment with MinecraftEdu. *Int. J. Educ. Technol. High. Educ.* **2016**, *13*, 1–11. [CrossRef]
34. Hiltz, S.R.; Coppola, N.; Rotter, N.; Turoff, M.; Benbunan-Fich, R. Measuring the importance of collaborative learning for the effectiveness of ALN: A multi-measure, multi-method approach. *J. Asynchronous Learn. Netw.* **2000**, *4*, 103–125. Available online: <https://olj.onlinelearningconsortium.org/index.php/olj/article/view/1904> (accessed on 11 September 2020). [CrossRef]
35. Laros, F.J.M.; Steenkamp, J.B.E.M. Emotions in consumer behavior: A hierarchical approach. *J. Bus. Res.* **2005**, *58*, 1437–1445. [CrossRef]
36. Hair, J.F.; Anderson, R.; Tatham, R.L.; Black, W. *Multivariate Data Analysis*, 5th ed.; Prentice Hall: Upper Saddle River, NJ, USA, 1998.
37. Cohen, L.; Manion, L.; Morrison, K. *Research Methods in Education*; Routledge: New York, NY, USA, 2000.
38. Goetz, J.P.; Lecompte, M.D. *Etnografía y Diseño Cualitativo en Investigación Educativa. Evaluación del Diseño Etnográfico*; Ediciones Morata: Madrid, Spain, 1988.

39. York, R.O. *Statistics for Human Service Evaluation*; SAGE Publications: Los Angeles, CA, USA, 2016.
40. Cohen, J. *Statistical Power Analysis for the Behavioral Sciences*, 2nd ed.; Lawrence Erlbaum Associates, Inc.: Hillsdale, NJ, USA, 1988.
41. Tondeur, J.; van Braak, J.; Ertmer, P.A.; Ottenbreit-Leftwich, A. Understanding the relationship between teachers' pedagogical beliefs and technology use in education: A systematic review of qualitative evidence. *Educ. Technol. Res. Dev.* **2017**, *65*, 555–575. [[CrossRef](#)]
42. Gudmundsdottir, G.B.; Gassó, H.H.; Rubio, J.C.C.; Hatlevik, O.E. Student teachers' responsible use of ICT: Examining two samples in Spain and Norway. *Comput. Educ.* **2020**, *152*, 103877. [[CrossRef](#)]
43. Cárdenas, L.A.V.; Tavernise, A.; Bertacchini, F.; Gabriele, L.; Valenti, A.; Pantano, P.; Bilotta, E. An Educational Coding Laboratory for Elementary Pre-service Teachers: A Qualitative Approach. *Int. J. Eng. Pedagog.* **2016**, *6*, 11–17. [[CrossRef](#)]
44. Gabriele, L.; Bertacchini, F.; Tavernise, A.; Vaca-Cárdenas, L.; Pantano, P.; Bilotta, E. Lesson Planning by Computational Thinking Skills in Italian Pre-service Teachers. *Inform. Educ.* **2019**, *18*, 69–104. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).