

Cite as: Moreno-León, J., Robles, G., & Román-González, M. (2016). Code to learn: Where does it belong in the K-12 curriculum? *Journal of Information Technology Education: Research*, 15, 283-303. Retrieved from <http://www.informingscience.org/Publications/3521>

Code to Learn: Where Does It Belong in the K-12 Curriculum?

Jesús Moreno-León
Programamos.es & Universidad Rey Juan Carlos,
Sevilla, Spain

jesus.moreno@programamos.es

Gregorio Robles
Universidad Rey Juan Carlos,
Madrid, Spain

grex@gsyc.urjc.es

Marcos Román-González
Universidad Nacional de
Educación a Distancia,
Madrid, Spain

mroman@edu.uned.es

Abstract

The introduction of computer programming in K-12 has become mainstream in the last years, as countries around the world are making coding part of their curriculum. Nevertheless, there is a lack of empirical studies that investigate how learning to program at an early age affects other school subjects. In this regard, this paper compares three quasi-experimental research designs conducted in three different schools (n=129 students from 2nd and 6th grade), in order to assess the impact of introducing programming with Scratch at different stages and in several subjects. While both 6th grade experimental groups working with coding activities showed a statistically significant improvement in terms of academic performance, this was not the case in the 2nd grade classroom. Notable disparity was also found regarding the subject in which the programming activities were included, as in social studies the effect size was double that in mathematics.

Keywords: elementary education, improving classroom teaching, interdisciplinary projects, programming and programming languages, teaching/learning strategies, computer education, Scratch.

Introduction

Computer programming is becoming a basic skill that allows participating effectively in a world

Material published as part of this publication, either on-line or in print, is copyrighted by the Informing Science Institute. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission and payment of a fee. Contact Publisher@InformingScience.org to request redistribution permission.

full of digital objects (González, 2014), and in recent years we are witnessing initiatives worldwide to promote the teaching of programming in schools. However, there is little consensus about the way programming should be included in the curriculum (Grover & Pea, 2013).

The European Schoolnet has elaborated a report where 20 European Ministries

Editor: Keith A. Willoughby

Submitted: November 8, 2015; Revised: May 16, 2016; Accepted: June 9, 2016

of Education give an overview of their current initiatives and plans regarding computer programming (Balanskat & Engelhardt, 2015). There are countries where programming is an end in itself, as the focus is on the potential benefits for those who know to program when they enter the labor market. This approach is based on studies that estimate that in the near future there will be a significant shortage of well-trained professionals in this field in Europe (Gareis et al., 2014) and in the US (Cuny, 2012). Some countries have decided to create a new specific subject, while others have included coding in the general ICT/technology course. Other governments are considering programming from a different perspective, as they pay more attention to the educational impact of learning to program at an early age, using coding as a medium to learn other disciplines, an approach summarized by Mitchel Resnick (2013) in *Learn to code, code to learn*.

In addition, there is also disagreement about the educational level in which coding should be introduced for the first time, as there are countries that begin in primary education, such as Estonia or Israel, while others delay it until secondary, as Austria or Ireland, or vocational stages, as is the case in Czech Republic. The nature of the education may differ as well; in some cases it is a mandatory subject, such as in England or Slovakia, others an optional one, as in Lithuania or Poland. The absence of a common way to proceed is partially related to a lack of empirical research on the topic that could provide evidence to policy makers (Grover & Pea, 2013). An example that illustrates the need for investigations in this area is the special theme of the 2015 Workshop in Primary and Secondary Computing Education: “*Special Theme: Computing? How young is too young?*”, which was chosen since there are relatively few results published on early computing education, with the aim that the workshop would contribute original research on this topic, including teaching approaches.

The research question that this paper addresses is whether **the use of programming with Scratch as an educational tool accelerates the learning curve, in terms of academic performance, in primary education (RQ1)**.

In primary education there are students both in the childhood stage (7-8 years old) and in early adolescence (11-12 years old). As other authors have highlighted, it is very relevant to compare the differential cognitive responses of these two age groups, as the brain structure and functions undergo a very significant maturation throughout these years (Huttenlocher & Dabholkar, 1997; Lezak, 2004). More specifically, the prefrontal neurological systems are still immature in the younger age group, while they have already emerged in the older one. In terms of Piaget (2003), it marks the transition from the stage of “concrete operations” to the stage of “formal operations”.

In addition, school subjects differ both in terms of cognitive complexity and students’ motivation. In this regard, since the 1990s several researchers have studied how cognition and motivation must interact to create optimal conditions for learning and academic performance. Thus, Núñez Pérez et al. (1998) address this issue specifically in learners between 9 and 13 years, and Boekaerts (1996) highlights the importance of perceived capacity for the successful completion of academic tasks. This research shows that the active involvement of students in the learning process increases when they feel competent, that is, when they trust their own abilities and have high expectations of self-efficacy, when they value the work, and when they feel responsible for the learning objectives (Zimmerman, Bandura, & Martinez-Pons, 1992). This influences both the cognitive and metacognitive strategies that are triggered when students perform tasks, as the regulation of effort and persistence, which in turn has a direct and positive effect on academic performance (Núñez Pérez et al., 1998).

In consequence, our main research question derives in **two specific questions**:

- **RQ1.1. Is there a differential effect depending on the academic year in which programming with Scratch is included?**

- **RQ1.2. Is there a differential effect depending on the school subject where programming with Scratch is integrated?**

Aiming to address these questions, we have performed an investigation in three schools during the academic year 2014-2015, where we conducted three quasi-experimental research designs involving 129 students from 2nd and 6th grade to assess and compare the impact of introducing programming with Scratch at different stages and in several subjects. A review of literature on the educational use of computer programming is presented in the Background section. In the Methodology section we justify the election of Scratch as the programming language for the investigation, describe each of the phases in which the work with teachers and students was divided, and outline the methods and instruments utilized for assessment. The findings of the quasi-experiments are presented in the Results section, where we evaluate the academic performance of control and experimental groups and compare the effect size in each school to assess the differential impact on the learning curve of participants, measure the computational thinking and complexity of the projects programmed by teachers and students, and analyze the answers of the satisfaction surveys. Based on these results, the research questions are addressed in the Discussion section, where we also present some threats to validity of the study. Finally, in the Conclusion we summarize the major ideas of our work, raise some possible implications, and present ideas for further research.

Background

Even if the issues addressed in this paper are current, the educational use of programming is not new. The Logo programming language was created back in the 1960s with the aim of introducing children to the use of computers as instruments for learning and for enhancing creativity (Feurzeig, 2010; Papert, 1980). Logo was “designed to provide an environment in which learners determine a problem to solve, make choices, experiment, try out solutions, and build on what they already know” (Watt, 1982). In the following decades, the impact of programming with Logo on learning other subjects was intensively investigated, as stated in several literature reviews written in the 1990s (Clements & Meredith, 1993; Clements & Sarama, 1997; Palumbo, 1990).

Focusing on mathematics and primary education, research shows improvements at classifying figures (Battista & Clements, 1988) and enhancements on geometric thinking (Johnson-Gentile, Clements, & Battista, 1994). In an investigation working with angles, the learners showed more generalized and mathematically oriented conceptualizations (Clements & Battista, 1989). Regarding competencies in linear measurement, a study showed evidence that students who used Logo were more accurate (P. F. Campbell, 1987). Nonetheless, there are also investigations that did not observe benefits in the learning of students after interventions to teach mathematics with this programming language, as Olive (1991) or Hamada (1987) reported.

But the investigations on Logo were not focused only on math. Some scholars studied the use of this programming language and its impact on language arts, demonstrating that it fostered conversational interactions between children (Genishi, McCollum, & Strand, 1985), increased perceptual-language skills (Lehrer & DeBernard, 1987), and enhanced scores on vocabulary and listening comprehension in the Stanford Achievement Test (Robinson, Gilley, & Uhlig, 1988). However, the impact of programming on language arts was not deeply investigated, and, hence, there is need for research to extend and explain the reported positive findings (Clements & Sarama, 1997).

In late 1990s and early 2000s the interest on programming in K-12 declined (De Raadt, 2004). However in the last years new studies have been performed on the educational impact of programming using new visual programming languages based on blocks (Weintrop & Wilensky,

2015), the most notable example being *Scratch* (Maloney, Resnick, Rusk, Silverman, & Eastmond, 2010), which try to address one of the major issues related to teaching introductory programming courses: the amount of time spent on the language's syntax (Al-Imamy, Alizadeh, & Nour, 2006). In the development of *Scratch*, creators were inspired and informed by the success and posterior decline in the use of *Logo* and other previous programming languages, aiming to make *Scratch* more meaningful for learners, more *tinkerable*, and more social (Resnick et al., 2009). Although a majority of recent investigations are focused on the development of computational thinking (Lye & Koh, 2014), there is relevant research that analyzes the effects of coding in the learning of other subjects.

A positive correlation between programming quizzes and better math test scores has been detected in a 5th grade class (Lewis & Shah, 2012); there is evidence that 3rd graders improve in the identification and comparison of numbers (Zavala, Gallardo, & García-Ruíz, 2013); a significantly more positive attitude towards math after including programming activities in middle school has been proved (Ke, 2014); a significant improvement on middle school students' learning of probability concepts after developing *Scratch* video games has been reported (Akpınar & Aslan, 2015); and a significant increase in mathematical thinking in 6th grade students after the use of *Scratch* has been found (Calao, Moreno-León, Correa, & Robles, 2015).

There is as well recent research on the use of programming and language arts. Sanjanaashree, Kumar and Soman (2014) describe how programming with *Scratch* can be used "to learn single sentence construction of secondary language (English) through primary language (Tamil)"; Burke and Kafai (2010) run an after-school creative writing workshop for secondary students and showed that 60% of participants pointed out that their storytelling skills had improved; and Burke (2012) concludes that coding with *Scratch* is a framework that facilitates digital composition of students in middle school.

Finally, there are studies focused on the use of programming in science lessons using *Logo* (Kafai, Ching, & Marshall, 1997; Yarnall & Kafai, 1996), *Netlogo* (Dickes & Sengupta, 2013; Wilensky & Reisman, 2006), and *Scratch* (Lai & Lai, 2012). To the best knowledge of the authors, no specific research on the effect of coding in social studies exists.

Although recent investigations have a very promising outlook, most of the reviewed papers do not follow basic recommendations to develop research in education (L. Cohen, Manion, & Morrison, 2007). It is therefore necessary to develop more empirical studies using new visual programming languages, providing quantitative data, using control and experimental groups of students with similar characteristics, and performing pre- and post-tests.

Methodology

The programming language used in this study has been *Scratch*, as it allows the creation of many different types of projects, so that learners with diverse interests and learning styles can find ways to express themselves through programming (Resnick et al., 2009). Furthermore, *Scratch* creators claim that when children learn to program with this application, they develop "design strategies (such as modularization and iterative design) that carry over to non-programming domains" (Resnick et al., 2009).

The study was divided into two phases. First, the participating teachers received a four-week training course in which they learned how to create applications with *Scratch* and several strategies to use programming as an educational tool in their specific subjects. The course syllabus was as follows:

1. Introduction to programming with Scratch: the Scratch website, sequence of instructions, control structures, my first video game.
2. Project-Based-Learning (PBL) with Scratch: parallelism and events, testing and debugging, using PBL with Scratch in your class.
3. Using data in Scratch projects: variables and lists.
4. Hands-on! Development of educational video games, designing programming activities for your lessons.

The course was developed using a teaching online community, Inevery Crea, where educators had access to videos, articles, and other resources, as well as to several communications tools. A total of 15 teachers (from Spain and Argentina) participated in the training phase, generating 58 comments and 48 private messages.

The second phase consisted in the development of a quasi-experiment in each participating school. In order to perform the study, teachers were encouraged to create a control group and an experimental group with students of similar characteristics in regard to age, academic performance, gender distribution, and number of pupils. This was a fundamental requirement that some schools were not able to meet, which highlights the difficulties for carrying out educational investigations with quasi-experimental designs: some schools only had one group in each grade; others had organized groups by performance, separating high and low achieving students; there were schools where the subject involved in the investigation was taught by different teachers in each of the groups of that grade; finally, in some cases school policy avoided the possibility of using different resources and methodologies in groups of the same level.

Once groups were established, teachers designed a pre-test to assess the knowledge that students had of the unit they were about to start working on at school. All students, both from the control and experimental group, took the pre-test. Then, the control group started working the unit using the same educational methodology and tools they had been using, while the experimental group included programming activities to work the contents of the unit. In each school, control and experimental groups were taught by the same teacher, although the lessons were imparted separately, in different classrooms. Finally, after at least four weeks of classes, all students took a post-test to assess their improvement. A common test could not be used in all the schools, as it was adapted to the specific contents of the unit, but each participating school used the same instrument for pre-test and post-test. The tests can be found in the replication package of the paper (<http://gsyc.urjc.es/~grex/repro/2016-jite-code-to-learn>).

In this paper we present the results of the three schools that met the requirements for a quasi-experimental research design, and afterwards compare and discuss them in terms of academic year and school subject. Hence, our approach combines quasi-experimental and causal-comparative methods, as indicated by Schenker and Rumrill (2004).

Table 1 shows the characteristics of the investigation in each participating school in terms of grade in which the work was carried out, the number of students in the control and experimental groups, the involved subject and the specific contents of the unit, as well as the types of programming activities that were developed by the experimental groups.

Figure 1 presents one of the Scratch projects developed by students of the Corazón de María School, in which the character is asking the user for data to draw an angle.

Table 1. Characteristics of the quasi-experimental research design in each participating school.

School	Grade	n (control)	n (exp.)	Subject	Unit	Activities
Corazón de María	6 th	26	25	Math	Angles	Projects to draw, classify and perform operations on angles
La Jota	6 th	25	24	Social studies	European Union	Question-and-answer video games
La Inmaculada	2 nd	15	14	Language arts	Narrative structure	Storytelling projects

**Figure 1. Screenshot of a Scratch project developed by students participating in the investigation. The character in the figure is asking the user to input the degrees of an angle to be drawn on the screen.**

In addition to the pre- and post-test, the experimental groups filled a satisfaction survey at the end of the intervention. This survey was made up of 11 Likert items (5-point scale from “1=Strongly Disagree” to “5=Strongly Agree”), grouped into three sections, and an open question:

- Section 1. After working with Scratch during these weeks, do you agree with the following statements?
 1. I enjoyed the classes in which we have worked with Scratch.
 2. I found the classes in which we have worked with Scratch interesting.
 3. I found it difficult to work with Scratch
 4. I have learned a lot working with Scratch.
 5. Working with Scratch helps to learn about teamwork.
- Section 2. After working other subjects through programming, do you agree with the following statements?
 6. Scratch activities have helped me to learn more in other subjects.
 7. Working with Scratch encourages me to want to learn more about other subjects.
 8. Working with Scratch has encouraged me to make contact with other children from around the world through the Internet.
 9. Working with Scratch has encouraged me to learn on my own, looking for information on the Internet.
 10. After working with Scratch, I think it is very necessary to learn on my own how to look for information on the Internet.

- Section 3. Thinking about the future.
 11. In the future, I would like to work on something related to computers and technology.
- Open question. Would you like to comment something about the experience of working with Scratch during these weeks?

Finally, *Dr. Scratch* was used in order to measure the development of Computational Thinking (CT) demonstrated in the development of the projects (Moreno-León & Robles, 2015).

Dr. Scratch is a web tool based on Hairball (Boe et al., 2013) that implements several CT assessment methods found in the research literature (Brennan & Resnick, 2012; Moreno & Robles, 2014; Seiter & Foreman, 2013; Wilson, Hainey, & Connolly, 2012; Wolz, Hallberg, & Taylor, 2011). The CT score that *Dr. Scratch* assigns to projects is based on the degree of development of following dimensions of the CT competence: abstraction and problem decomposition, logical thinking, synchronization, parallelism, algorithmic notions of flow control, data representation, and user interactivity. These dimensions are evaluated by statically inspecting the source code of the project. Each dimension is assessed with a score from 0 to 3 points. The overall CT score is calculated by summing up the partial scores and, therefore, ranges from 0 to 21 points (Moreno-León, Robles, & Román-González, 2015).

Results

For our statistical decisions, we have set a 99% confidence level ($\alpha=0.01$).

Descriptive Statistics

Table 2 shows the results of the pre- and post-tests for the control and the experimental groups, including mean and standard deviation. If the results of the groups before the intervention are compared, we can see that the mean and the standard deviation are uniform, which is congruent with the fact that groups were formed by students of similar characteristics. However, the results after the intervention show a difference in favor to the experimental group in all three cases, although the differences vary from school to school. Figure 2 illustrates the results at the Corazón de María school, showing that groups departed from a very similar point and enhanced their mean scores along the investigation. However, the experimental group had a bigger improvement. In a similar way, Figure 3 shows that the difference between groups was more remarkable at La Jota school. The difference in favor to the experimental group is much smaller at La Inmaculada school, as shown in Figure 4.

Table 2. Pre- and post-tests results of control and experimental groups

		Control Group		Experimental Group	
		Pre-test	Post-test	Pre-test	Post-test
		(N=26)		(N=25)	
Corazón de María 6 th grade	Mean	4.73	6.27	5.12	7.48
	SD	1.25	1.19	1.51	1.42
		(N=25)		(N=24)	
La Jota 6 th grade	Mean	1.44	5.32	1.63	6.71
	SD	0.96	1.70	1.06	1.78
		(N=15)		(N=14)	
La Inmaculada 2 nd grade	Mean	4.93	6.33	5.00	6.50
	SD	1.39	1.29	1.75	1.74

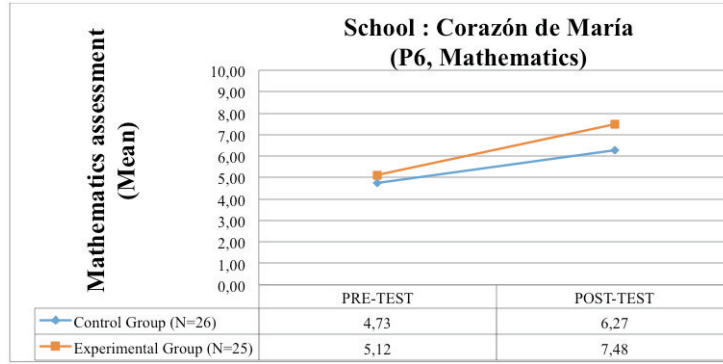


Figure 2. Corazón de María school. Comparing the improvement between pre- and post-tests of control and experimental groups

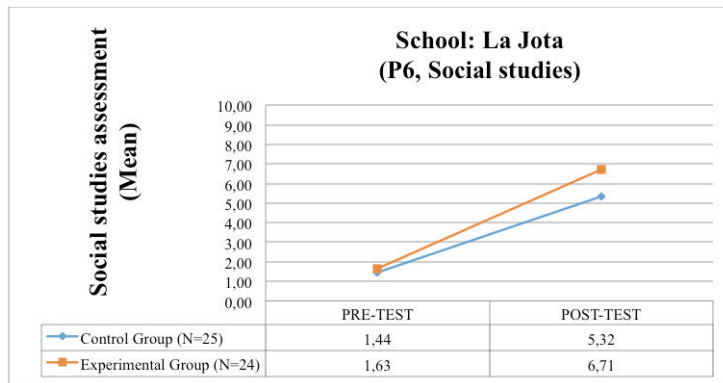


Figure 3. La Jota school. Comparing the improvement between pre- and post-tests of control and experimental groups

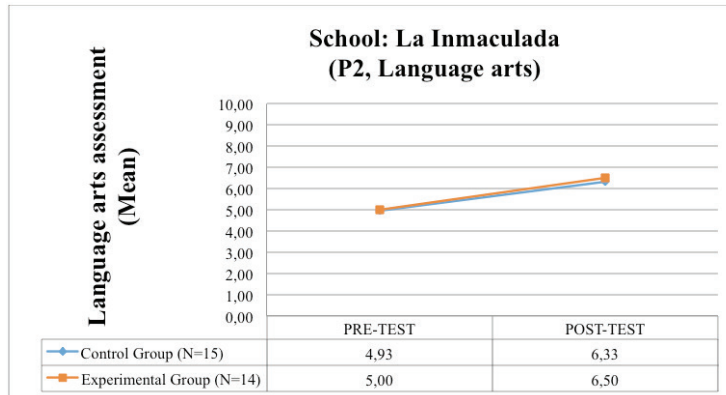


Figure 4. La Inmaculada school. Comparing the improvement between pre- and post-tests of control and experimental groups

T-test for Equal Means

To be able to prove that the bigger improvements of the experimental groups are statistically significant, the means of the groups under study have to be statistically analyzed by performing a t-test for independent samples. Thus, as a requirement of the t-test, the normality of the samples is verified using the Kolmogorov-Smirnov test. For all samples $p(Z_{k-s})$ is greater than 0.01, which means that their normality is accepted and the t-test can therefore be applied to test differences

between the control and the experimental group, both in pre- and post-test conditions for all schools.

Consequently, we used t-test for independent samples; results are presented in Table 3. If the pre-test conditions are analyzed, $p(F)$ is greater than 0.01 in all schools, which implies that the homoscedasticity assumption is met. Both at Corazón de María school and at La Jota school, there are no significant differences between control and experimental groups in pre-test condition [respectively $p(t) = 0.320 > 0.01$; $p(t) = 0.524 > 0.01$], but significant differences in post-test condition favoring the experimental group [respectively $p(t) = 0.002 < 0.01$; $p(t) = 0.008 < 0.01$]. Nevertheless, at La Inmaculada school there are no significant differences between the control and the experimental group neither in pre-test [$p(t) = 0.910 > 0.01$] nor in post-test [$p(t) = 0.771 > 0.01$].

Table 3. t-test for independent samples in the pre-test and post-test

	T-test for equal mean						
	t	df	Sig. (bilat)	M diff.	SE diff.	95% CI	
						Lower	Upper
Corazón de María							
Pre-test	-1.00	49	.32	-.39	.39	-1.17	.39
Post-test	-3.31	49	.002	-1.21	.37	-1.94	-.48
La Jota							
Pre-test	-.64	47	.52	-.19	.29	-.76	.39
Post-test	-2.79	47	.008	-1.39	.50	-2.39	-.39
La Inmaculada							
Pre-test	-.11	27	.91	-.07	.59	-1.27	1.13
Post-test	-.29	27	.77	-.17	.57	-1.33	1.00

Aiming to check differences between pre- and post-test at each school, both for control and experimental groups, a t-test for paired samples has been used. The improvement between pre- and post-tests for all groups and schools is statistically significant, as $p(t) < 0.01$ in all cases.

In summary, as expected, control and experimental groups had significant improvements between pre-test and post-test conditions in every participating school. Nevertheless, the enhancements of the experimental groups at the Corazón de María and La Jota schools were significantly higher. On the contrary, at La Inmaculada school the improvement has been statistically equal in both groups.

Effect Sizes

The overall effect size of the experiment is a good indicator to assess the real impact on the learning curve of participants. It is calculated for each school by the d of Cohen (J. Cohen, 1990).

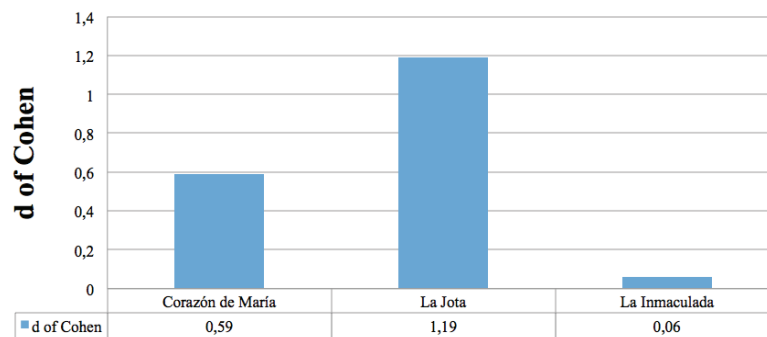


Figure 5: Effect size

Figure 5 shows the differences in the effect sizes, which are congruent with the results of the t-tests. At Corazón de María school the overall effect size of the experiment is $d = 0.59$, which can be considered as a moderate effect (Ellis, 2010). At La Jota school the overall effect size is $d = 1.19$, which is considered a large effect. Finally, at La Inmaculada school the overall effect size is $d = 0.06$, which can be considered as a null effect.

Analysis of Covariance (ANCOVA)

The Analysis of Covariance (ANCOVA) (Todman & Dugard, 2007) can contrast the differences in the dependent variable (in our case the post-test scores) according to a fixed factor (in our case, “control condition” or “experimental condition”) controlling statistically possible baseline differences between both groups (i.e., controlling any initial differences in the pre-test scores).

Table 4 shows a summary of the results of ANCOVA for each of the schools studied. At Corazón de María school [$p(\text{Fcondition}) = 0.001 < 0.01$] and at La Jota school [$p(\text{Fcondition}) = 0.008 < 0.01$] significant differences in the post-test between the control and the experimental groups can be stated, after having controlled statistically potential baseline differences between the groups. However, at La Inmaculada school this is not the case, as $p(\text{Fcondition}) = 0.771 > 0.01$, and no significant differences in the post-test between the groups can therefore be stated. Thus, the ANCOVA technique supports the findings with t-tests.

Table 4. ANCOVA. Tests of Between-Subjects Effects. Dependent Variable: Post-Test

Condition	Type III Sum of Squares	df	Mean square	F
Corazón de María	11.906	1	11.906	11.575**
La Jota	18.485	1	18.485	7.790**
La Inmaculada	.107	1	.107	.086

** $p(\text{F}) < 0.01$

Analysis of the Projects Developed by the Students

Scratch projects developed by the participating students from 6th grade experimental groups, which can be found in the replication package, were analyzed to discover if there were any differences in the complexity of the programs and the types of blocks used.

Figure 6 shows the results of the analysis with Dr. Scratch, including the mean score for each CT dimension (from 0 to 3) and the total CT score (in the range 0 to 21). The mean of the Scratch projects coded at Corazón de María school is higher for all computational thinking aspects (except for “user interactivity”) than those programmed at La Jota, and in consequence, the total CT score is also higher for projects developed at Corazón de María school. We performed t-test for independent samples in order to test the significance of the differences between the two schools. As a result, in spite of the small size of the sample (9 and 21 projects respectively), significant differences were found in the “abstraction” dimension [$p(t) = 0.000 < 0.01$] and in the total CT score [$p(t) = 0.001 < 0.01$].

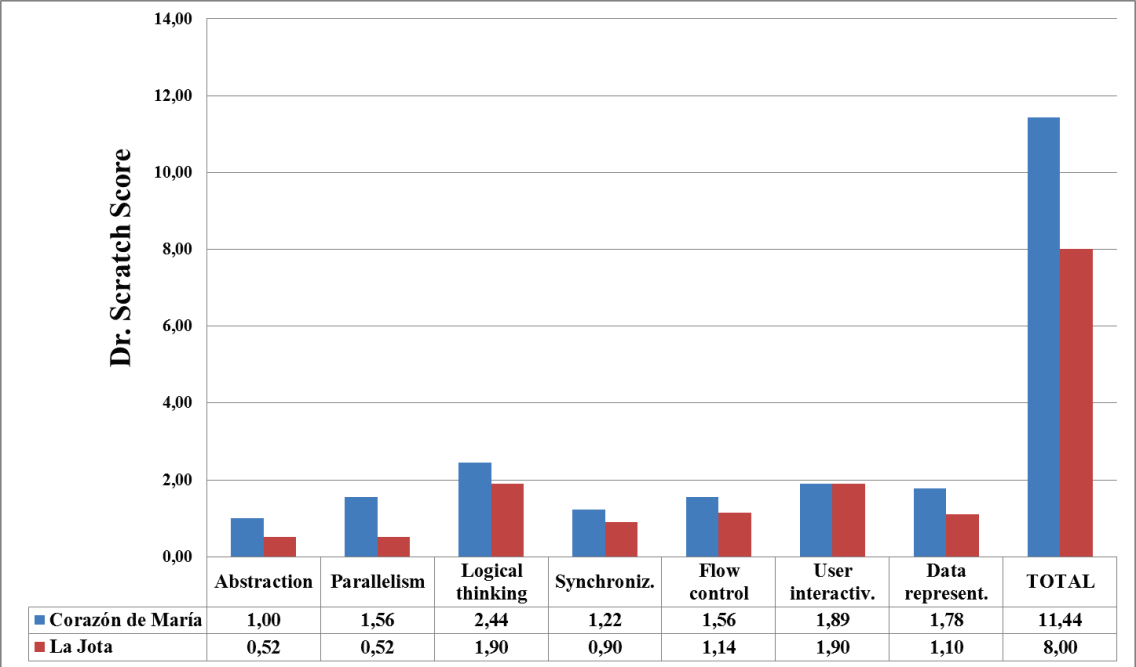


Figure 6. Analysis of the Scratch projects with Dr. Scratch:
Degree of development of different aspects of Computational Thinking

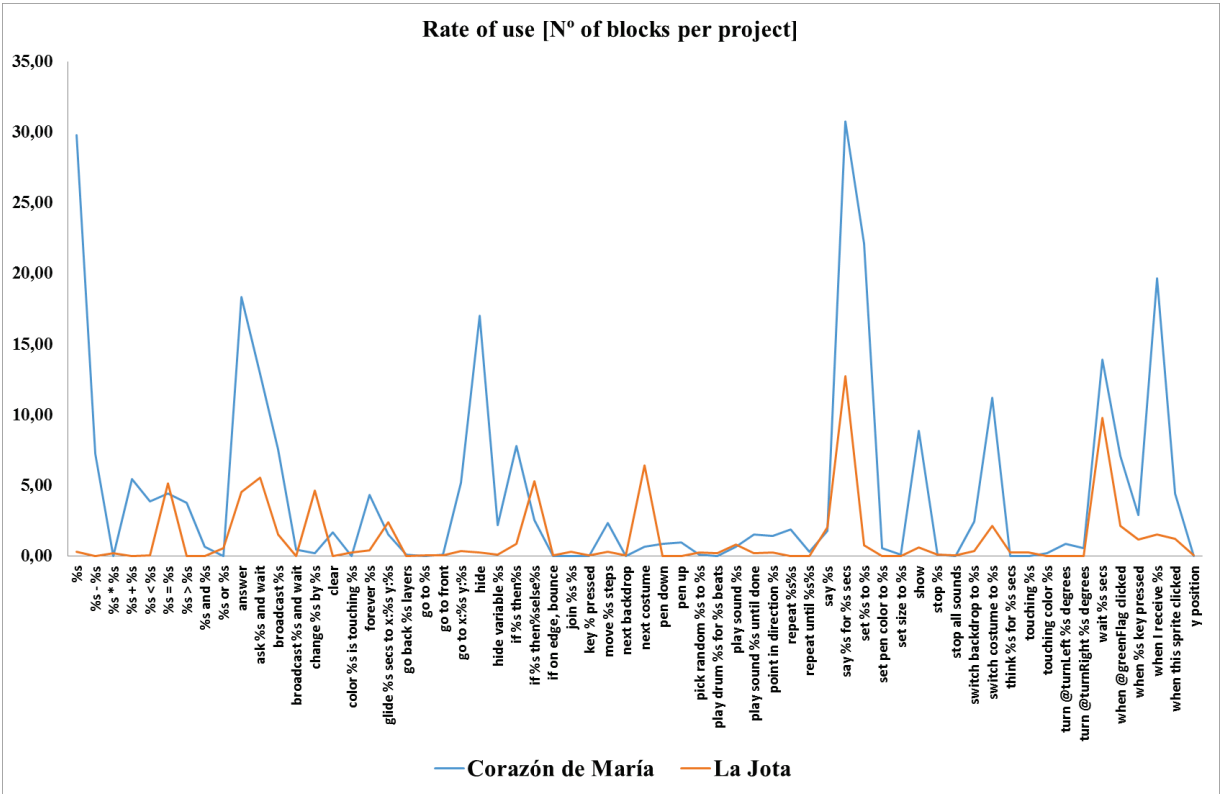


Figure 7. Rate of use of Scratch blocks in the projects of the students at each school

A higher development of CT implies higher levels of complexity in the projects and is also reflected in the type of *Scratch* blocks (equivalent to the instructions of traditional programming languages) used. As can be seen in Figure 7, projects developed at Corazón de María present a higher complexity, with a higher use rate for almost all types of blocks. These differences can be explained by studying the type of projects the students worked on. Although most of the projects from both schools are question-and-answer games, in those developed at Corazón de María (the ones related to angles) the characters perform arithmetic and logic operations to check the correctness of the answers, use pen blocks to draw the angles, and utilize motion blocks to make the sprites turn. In addition, users are allowed to choose the questions by clicking on different objects in the games of this school, which has an impact on the type of control and events blocks that are used.

Analysis of the Projects Developed by the Teachers

A different teacher at each school carried out the work with the students. Hence, we also compare the programming skills reached by each of the participating teachers in order to see if a relationship exists between the programming level of teachers and the outcome of their students.

At the end of the training phase, teachers were encouraged to develop and share a non-trivial *Scratch* project. We analyzed these projects, which can be found in the replication package, with the *Dr. Scratch* tool. Results indicate that the three teachers reached a very similar point of development, as the CT scores provided by *Dr. Scratch* were close:

- Teacher from Corazón de María, CT score in *Dr. Scratch*: 13/21
- Teacher from La Jota, CT score in *Dr. Scratch*: 14/21
- Teacher from La Inmaculada, CT score in *Dr. Scratch*: 12/21

From these results we infer that the differences we found in academic performance by students are to be attributed to causes other than the CT developed by their teachers.

Satisfaction Survey Results

The survey was filled out by all experimental groups, but only the responses of the 6th grade groups, at Corazón de María and La Jota schools, are being analyzed since they show sufficient reliability in their answers (Cronbach's alpha approximately equals to 0.80), a condition not met for the 2nd grade group. Responses to the survey from all groups and their analysis can be found in the replication package. Table 5 compares the mean for each question in the survey, for each of the groups separately, and gives the value of $p(t)$ for each mean difference between the experimental groups. The most important results are:

- Students from La Jota enjoyed working with *Scratch* (item 1) to a greater extent than students from Corazón de María (statistically significant). We extract some words from the answers to the open question (translated from Spanish) that point in this direction:
 - Corazón de María: “a bit boring”, “super boring”, “so boring”, “not fun”, “not much fun”, “kind of boring”.
 - La Jota: “lots of fun”, “very fun”, “interesting and fun”, “really fun”, “enjoyed it”.
- Students from Corazón de María found working with *Scratch* (item 3) more difficult than students from La Jota (statistically very significant).

- Corazón de María: “too hard”, “hard”, “a bit hard”, “complicated”.
- La Jota: no appearance of the words “hard” or “complicated” on any of the answers.
- Furthermore, working with *Scratch* has encouraged students from La Jota to learn to look for information on the Internet on their own (item 9) to a greater extent than students from Corazón de María (statistically very significant).
 - Corazón de María: No answers talking about this topic.
 - La Jota: “search information, research and try new things”.
- Finally, students from La Jota would like to work in the future on something related with computers & technology (item 11) to greater extent than students from Corazón de María (statistically significant).

Lastly, correlations between the items in which we found significant differences between the two schools have been calculated (Table 6). On the one hand, we found a very significant positive correlation between the enjoyment with *Scratch*, the motivation to seek information by oneself on the Internet, and the desire to engage in future tasks related to computers and technology. On the other hand, we found a significant negative correlation between the difficulty of working with *Scratch*, and the desire to engage in future tasks related to computers and technology.

Table 5: Satisfaction survey responses, comparison of means.

Section	Item	Overall mean (n=47)	Mean - Corazón de María (n=24)	Mean - La Jota (n=23)	p(t)
1 - Working with Scratch	1. Enjoyment	4.28	4.00	4.57	.035*
	2. Interest	4.17	4.08	4.26	.589
	3. Difficulty	2.68	3.25	2.09	.004**
	4. Learning	4.13	3.92	4.35	.147
	5. Teamwork	3.91	3.75	4.09	.295
2 - Learning other subjects through programming	6. Other subjects (learning)	3.57	3.63	3.52	.755
	7. Other subjects (motivation)	3.47	3.25	3.70	.218
	8. e-contact other children	2.74	2.50	3.00	.167
	9. Learning on my own (motiv.)	3.54	3.00	4.14	.005**
	10. Learning on my own (need)	3.72	3.42	4.05	.050
3 - Future	11. Future related to computers & technology	2.85	2.46	3.26	.028*

*p (t) < 0.05. **p (t) < 0.01

These results show notable differences in the motivation towards the programming assignments of the students from the two schools. While learners from La Jota seemed to enjoy the coding activities and the type of tasks that those activities implied, such as looking for information on the Internet, students from Corazón de María reported a less joyful attitude towards them.

Table 6: Correlations between questions of the survey. Cells with * indicate significant correlation $p(r) < 0.05$. Cells with ** indicate very significant correlation $p(r) < 0.01$.

	Item 1	Item 3	Item 9	Item 11
Item 1 - Enjoyment	1	-.195	.420**	.388**
Item 3 - Difficulty		1	-.058	-.290*
Item 9 - Learning on my own (motivation)			1	.148
Item 11 - Future related to computers & technology				1

* $p(r) < 0.05$. ** $p(r) < 0.01$

Discussion

In this section we will answer the research questions that this paper targets, discuss implications and possibilities for future research, and point out the threats to the validity of our results.

RQ1.1. Is there a differential effect depending on the academic year in which programming with Scratch is included? (First stage of primary education vs late stage of primary education)

The results indicate that the inclusion of computer programming with Scratch in the late primary education (6th grade) significantly accelerated the learning curve. However, this was not the case in 2nd grade.

One possible explanation for this differential effect may lie in the process of cognitive transfer in terms of age (Jaeggi, Buschkuhl, Shah, & Jonides, 2014; Karbach & Kray, 2009; Schiff & Vakil, 2015). Transfer is a cognitive process where the knowledge acquired in one situation is applied in a different context. To make transfer possible, learners must recognize the similarities between the two situations, activate the mindset acquired in the original context, and apply its common principles to the new context, ignoring irrelevant differences between the two situations (Chen, 1999). It is therefore a cognitive process that involves a high level of abstraction. In our research, students have to perform transfer between the learning achieved in a programming environment, an original new environment, to a situation of academic evaluation, a traditional exam situation. Based on the aforementioned literature, it seems plausible to interpret that older learners can achieve such transfer more effectively.

In this sense our results are in line with an investigation in which 6th graders outperform 3rd graders in transferring a cognitively learned skill using the Tower of Hanoi Puzzle to another domain (Schiff & Vakil, 2015). The resolution of the Tower of Hanoi Puzzle involves cognitive skills such as planning, logical, abstract and recursive thinking, and mental visualization-modeling, as well as self-monitoring and self-error correction skills (Guevara, Martínez, Aguirre, & González, 2012), all of which are also present in a computer programming environment like the one of Scratch.

Lastly, it is worth noting that although the learning curve was not accelerated in the 2nd grade group, it did not suffer a deceleration. This means that even though the use of programming with Scratch does not result in academic performance improvements in the first stages of primary education, it may be advisable to start introducing this resource at that age, as children can develop their coding and computational thinking skills without compromising other academic outcomes.

RQ1.2. Is there a differential effect depending on the school subject where programming with Scratch is integrated?

Although in both 6th grade groups the use of computer programming accelerated the learning curve, the effect size was two times larger in La Jota ($d=1.19$), where the investigation was in a social studies subject, than in Corazón de María ($d=0.59$), where programming was included in the math lessons.

Based on the differences found when analyzing the projects in both schools, we can state that the integration of programming with Scratch was more cognitively demanding, in terms of computational thinking, at Corazón de María, which is consistent with the higher cognitive complexity of mathematics compared to social studies at these ages. The cognitive demand of the students at La Jota was lower, but according to the results of the satisfaction surveys, it translated into greater enjoyment and sense of mastery of the tool (greater perceived self-efficacy), which induced a greater sense of autonomous learning and enhanced motivation in regard to present and future programming-related tasks.

In this line, recent studies demonstrate the importance of the motivational factors for an effective transfer of cognitive skills acquired in one domain to another context (Jaeggi et al., 2014). It seems therefore plausible to interpret that in the social studies class concurred the best combination of cognitive and motivational factors, producing a major effect on the academic performance. The effect at the math group, although not that outstanding, was also notable and valuable.

From these results we can derive some important implications, as it may be advisable to begin with the use of programming with Scratch as a transversal educational tool in subjects and units in which the integration is simple and easy, such as social studies, as a first step for students to gain confidence and autonomy. Later, introducing these activities in more complex subjects such as mathematics could be tackled. In summary, when introducing computer programming in primary education, an optimal adjustment between cognitive and motivational factors has to be found.

Threats to Validity

This section discusses the different types of threats to the validity of our quasi-experiments, in order of decreasing priority: internal, external, construct and conclusion validity (D. T. Campbell & Stanley, 1963).

The difficulties for schools to meet the requirements for quasi-experimental designs present a clear threat to the internal validity of the study, as they affect the size and diversity of the sample.

An important threat to the internal validity is the role of the teachers, who voluntarily participated in the study. Although in the guidelines we encouraged them to design objective tests with simple answers in order to mitigate or minimize the potential bias in the marking process, both the design and the marking of the pre- and post-tests were performed by them, which could introduce bias.

Teachers may also have suffered from the “experimenter effect”, as they might have, even unconsciously, imparted lessons of the experimental groups with a higher motivation, willing to see the positive impact of programming.

External validity refers to the generalizability of the results of the experiment. In this regard, our sample is limited in size, being composed of only three participating schools. In our quasi-experiments, no socio-economic, gender nor other relevant characteristics of students and their environment have been considered, lessening the confidence of generalizing our results and requiring further research.

Construct validity concerns the way we defined our measurement and whether it measures the properties we really intend to capture. In this regard, the students involved in the investigation have probably experienced extra motivation, as this was their first interaction with computer programming. Further research should examine if the results are different when the participating students have prior coding experience. It has also to be considered that almost all measuring instruments (the academic performance tests and the satisfaction survey) have been designed *ad hoc* for this research, and have not gone through a validation process to make them standardized instruments. The case of *Dr. Scratch* is different, as the tool is currently in the process of a psychometric validation.

Conclusion validity relates to the degree to which conclusions about the relationship among variables, based on the data, are correct. These issues have been addressed in the design of the experiment.

Finally, the comparison of the results of the different quasi-experiments from a causal-comparative approach raises other threats to validity of the investigation. Thus, the main threat refers to the internal validity of the comparison, as two independent comparisons are tackled (one concerning variable “academic year”, another one related to variable “school subject”) not taking into consideration the possible interactions that can occur between them.

Conclusion

This paper presents a set of quasi-experiments that measures the educational impact of computer programming. It offers a valuable and complete perspective of a complex phenomenon by offering a triangular point of view with (1) academic performance data, (2) student perception and (3) the assessment of the projects with *Dr. Scratch*. By doing so, it aims to provide evidence to help policy makers and educators to take decisions on the stage and subjects in which these activities could be introduced in the curriculum. Therefore, using the same methodology and instruments, the investigation was carried out in three schools involving a 2nd grade language arts class, a 6th grade math class, and a 6th grade social studies class.

The results indicate that programming with *Scratch* accelerates the learning curve of the experimental groups in both 6th grade classes. Nevertheless the effect size was twice as big in the social studies than in math. In order to explain this difference, we analyzed the *Scratch* projects developed by the students and compared their answers to the satisfaction surveys. Our conclusion is that the bigger effect size is due to a better adjustment between cognitive and motivational factors, as learners in the social studies class showed higher levels of enjoyment and confidence in self-learning. These results open new questions that should be further investigated regarding the differences in the educational impact of computer programming depending on the type of subject or activity in which it is integrated.

The results with a 2nd grade class, where programming was introduced in language arts lessons, show that coding with *Scratch* does not accelerate the learning curve (but does not slow it either, even if programming is taught in addition to the actual content of the subject). Our conclusion is that differences between 2nd and 6th grade students are because the older students have a higher level of cognitive maturation that allows them to transfer learning from programming environments to other domains.

This work opens the possibility for future lines of research. On one hand, some of the discussed threats to validity could be addressed in other studies by introducing a more strict experimental design, e.g., having double blind controls, or by expanding the sample to become representative of the population and its characteristics (including possibly socio-economic, gender, and other relevant characteristics), in addition to other school subjects and ages. Future investigations

should include as well measuring instruments with accredited psychometric properties to ensure reliability and validity.

In the reviewed literature, it is common that researchers participate in the training of the students that take part in the study. In the investigation presented in this paper, this has been done completely by the participating teachers, after receiving an online four-week training course. These teachers have been able to successfully incorporate programming in their lessons and, by doing so, have improved the learning outcomes of their students. It should be noted that a majority of principals and superintendents of K-12 schools in the United States point out that the inability to hire or train teachers to lead computer science classes prevents schools and districts from offering them, as they consider teachers need extensive training (Google, 2015). Our observation should be confirmed in future research, but if proved to be valid, it could contribute to overcoming some of the barriers to extend computer programming and computer science education in schools.

References

- Akpinar, Y., & Aslan, Ü. (2015). Supporting children's learning of probability through video game programming. *Journal of Educational Computing Research*, 53(2), 228-259.
- Al-Imamy, S., Alizadeh, J., & Nour, M. A. (2006). On the development of a programming teaching tool: The effect of teaching by templates on the learning process. *Journal of Information Technology Education: Research*, 5, 271-283. Retrieved from <http://www.informingscience.org/Publications/247>
- Balanskat, A., & Engelhardt, K. (2015). *Computing our future: Computer programming and coding-priorities, school curricula and initiatives across Europe*. European Schoolnet. Retrieved from <http://www.eun.org/publications/detail?publicationID=661>
- Battista, M. T., & Clements, D. H. (1988). A case for a Logo-based elementary school geometry curriculum. *Arithmetic Teacher*, 36(3), 11-17.
- Boe, B., Hill, C., Len, M., Dreschler, G., Conrad, P., & Franklin, D. (2013). Hairball: Lint-inspired static analysis of scratch projects. *Proceeding of the 44th ACM technical symposium on Computer science education* (pp. 215–220). Denver, CO, USA: ACM. doi: 10.1145/2445196.2445265
- Boekaerts, M. (1996). Self-regulated learning at the junction of cognition and motivation. *European Psychologist*, 1(2), 100-112.
- Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Proceedings of the 2012 annual meeting of the American Educational Research Association*, Vancouver, Canada: AERA. Retrieved from http://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf
- Burke, Q. (2012). The markings of a new pencil: Introducing programming-as-writing in the middle school classroom. *The Journal of Media Literacy Education*, 4(2).
- Burke, Q., & Kafai, Y. B. (2010). Programming & storytelling: Opportunities for learning about coding & composition. *Proceedings of the 9th International Conference on Interaction Design and Children* (pp. 348–351). Barcelona, Spain: ACM. doi: 10.1145/1810543.1810611
- Calao, L. A., Moreno-León, J., Correa, H. E., & Robles, G. (2015). Developing mathematical thinking with Scratch: An experiment with 6th grade students. In *Design for Teaching and Learning in a Networked World, 10th European Conference on Technology Enhanced Learning, EC-TEL 2015* (pp. 17-27). Toledo, Spain: Springer. doi: 10.1007/978-3-319-24258-3_2
- Campbell, D. T., & Stanley, J. C. (1963). *Experimental and quasi-experimental designs for research on teaching*: American Educational Research Association.
- Campbell, P. F. (1987). Measuring distance: Children's use of number and unit. *Final report submitted to the National Institute of Mental Health Under the ADAMHA Small Grant Award Program*. Grant No. MSMA, 1, R03.

Code to Learn: Where Does It Belong in the K-12 Curriculum?

- Chen, Z. (1999). Schema induction in children's analogical problem solving. *Journal of Educational Psychology*, 91(4), 703.
- Clements, D. H., & Battista, M. T. (1989). Learning of geometric concepts in a Logo environment. *Journal for Research in Mathematics Education*, 20(5), 450-467.
- Clements, D. H., & Meredith, J. S. (1993). Research on Logo: Effects and efficacy. *Journal of Computing in Childhood Education*, 4(4), 263-290.
- Clements, D. H., & Sarama, J. (1997). Research on Logo: A decade of progress. *Computers in the Schools*, 14(1-2), 9-46.
- Cohen, J. (1990). Things I have learned (so far). *American Psychologist*, 45(12), 1304.
- Cohen, L., Manion, L., & Morrison, K. (2007). *Research methods in education*. New York: Routledge.
- Cuny, J. (2012). Transforming high school computing: A call to action. *ACM Inroads*, 3(2), 32-36.
- De Raadt, M. (2004). Searching for tomorrow's programmers. *Issues in Informing Science and Information Technology*, 1, 597-603.
- Dickes, A. C., & Sengupta, P. (2013). Learning natural selection in 4th grade with multi-agent-based computational models. *Research in Science Education*, 43(3), 921-953.
- Ellis, P. D. (2010). *The essential guide to effect sizes: Statistical power, meta-analysis, and the interpretation of research results*: Cambridge University Press.
- Feurzeig, W. (2010). Toward a culture of creativity: A personal perspective on Logo's early years and ongoing potential. *International Journal of Computers for Mathematical Learning*, 15(3), 257-265.
- Gareis, K., Husing, T., Birov, S., Bludova, I., Schulz, C., & Korte, W. B. (2014). *E-skills for jobs in Europe: Measuring progress and moving ahead*. European Commission. Retrieved from <http://eskills-monitor2013.eu/results>
- Genishi, C., McCollum, P., & Strand, E. (1985). Research currents: The interactional richness of children's computer use. *Language Arts*, 526-532.
- González, M. R. (2014). Aprender a programar 'apps' como enriquecimiento curricular en alumnado de alta capacidad [To learn programming 'apps' as curriculum enrichment on gifted students]. *Bordón. Revista de Pedagogía*, 66(4), 135-155.
- Google. (2015). *Searching for computer science: Access and barriers in U.S. K-12 education*. Retrieved from https://services.google.com/fh/files/misc/searching-for-computer-science_report.pdf
- Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher*, 42(1), 38-43.
- Guevara, M. A., Martínez, L. E. R., Aguirre, F. A. R., & González, M. H. (2012). Prefrontal-parietal correlation during performance of the towers of Hanoi task in male children, adolescents and young adults. *Developmental Cognitive Neuroscience*, 2(1), 129-138.
- Hamada, R. M. (1987). The relationship between learning Logo and proficiency in mathematics. *Dissertation Abstracts International*, 47, 2510-A.
- Huttenlocher, P. R., & Dabholkar, A. S. (1997). Regional differences in synaptogenesis in human cerebral cortex. *Journal of comparative Neurology*, 387(2), 167-178.
- Jaeggi, S. M., Buschkuhl, M., Shah, P., & Jonides, J. (2014). The role of individual differences in cognitive training and transfer. *Memory & Cognition*, 42(3), 464-480.
- Johnson-Gentile, K., Clements, D. H., & Battista, M. T. (1994). Effects of computer and noncomputer environments on students' conceptualizations of geometric motions. *Journal of Educational Computing Research*, 11(2), 121-140.
- Kafai, Y. B., Ching, C. C., & Marshall, S. (1997). Children as designers of educational multimedia software. *Computers & Education*, 29(2), 117-126.

- Karbach, J., & Kray, J. (2009). How useful is executive control training? Age differences in near and far transfer of task-switching training. *Developmental Science*, 12(6), 978-990.
- Ke, F. (2014). An implementation of design-based learning through creating educational computer games: A case study on mathematics learning during design and computing. *Computers & Education*, 73, 26-39.
- Lai, C.-S., & Lai, M.-H. (2012). Using computer programming to enhance science learning for 5th graders in Taipei. In *Computer, Consumer and Control (IS3C), 2012 International Symposium on* (pp. 146-148). Taichung, Taiwan: IEEE. doi: 10.1109/IS3C.2012.45
- Lehrer, R., & DeBernard, A. (1987). Language of learning and language of computing: The perceptual-language model. *Journal of Educational Psychology*, 79(1), 41.
- Lewis, C. M., & Shah, N. (2012). Building upon and enriching grade four mathematics standards with programming curriculum. *Proceedings of the 43rd ACM technical symposium on Computer Science Education* (pp. 57-62). Raleigh, NC, USA: ACM. doi: 10.1145/2157136.2157156
- Lezak, M. D. (2004). *Neuropsychological assessment*: Oxford University Press.
- Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51-61.
- Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The scratch programming language and environment. *ACM Transactions on Computing Education (TOCE)*, 10(4), 16.
- Moreno, J., & Robles, G. (2014). Automatic detection of bad programming habits in scratch: A preliminary study. *Proceedings of the Frontiers in Education Conference (FIE), 2014 IEEE* (pp. 1-4). Madrid, Spain: IEEE. doi: 10.1109/FIE.2014.7044055
- Moreno-León, J., & Robles, G. (2015). Analyze your Scratch projects with Dr. Scratch and assess your computational thinking skills. *Proceedings of the Scratch Conference 2015*, (pp. 48-53). Amsterdam, The Netherlands.
- Moreno-León, J., Robles G., & Román-González, M. (2015). Dr. Scratch: Automatic analysis of scratch projects to assess and foster computational thinking. *RED, Revista de Educación a Distancia*, 46. Retrieved from <http://www.um.es/ead/red/46>
- Núñez Pérez, J. C., González García, J. A., García Rodríguez, M. S., González-Pumariega Solís, S., Roces Montero, C., Álvarez Pérez, L., & González Torres, M. d. C. (1998). Estrategias de aprendizaje, auto-concepto y rendimiento académico [Learning strategies, self-concept and academic achievement]. *Psicothema*, 10(1) 97-109.
- Olive, J. (1991). LOGO programming and geometric understanding: An in-depth study. *Journal for Research in Mathematics Education*, 22(2), 90-111.
- Palumbo, D. B. (1990). Programming language/problem-solving research: A review of relevant issues. *Review of Educational Research*, 60(1), 65-89.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*: Basic Books, Inc.
- Piaget, J. (2003). Cognitive development in children: Piaget: Development and learning. *Journal of Research in Science Teaching*, 40, S8-S18.
- Resnick, M. (2013). Learn to code, code to learn. *EdSurge*, May. Retrieved from <https://www.edsurge.com/news/2013-05-08-learn-to-code-code-to-learn>
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., . . . Silverman, B. (2009). Scratch: programming for all. *Communications of the ACM*, 52(11), 60-67.
- Robinson, M. A., Gilley, W. F., & Uhlig, G. E. (1988). The effects of guided discovery Logo on SAT performance of first grade students. *Education*, 109(2).

Code to Learn: Where Does It Belong in the K-12 Curriculum?

- Sanjanaashree, P., Kumar, M. A., & Soman, K. (2014). Language learning for visual and auditory learners using scratch toolkit. *Proceedings of the Computer Communication and Informatics (ICCCI), 2014 International Conference on* (pp. 1-5). Coimbatore, India: IEEE. doi:10.1109/ICCCI.2014.6921765
- Schenker, J. D., & Rumrill Jr, P. D. (2004). Causal-comparative research designs. *Journal of Vocational Rehabilitation, 21*(3), 117-121.
- Schiff, R., & Vakil, E. (2015). Age differences in cognitive skill learning, retention and transfer: The case of the Tower of Hanoi Puzzle. *Learning and Individual Differences, 39*, 164-171.
- Seiter, L., & Foreman, B. (2013). Modeling the learning progressions of computational thinking of primary grade students. *Proceedings of the ninth annual international ACM conference on International computing education research* (pp. 59-66). Auckland, New Zealand: ACM. doi:10.1145/2493394.2493403
- Todman, J., & Dugard, P. (2007). *Approaching multivariate analysis: An introduction for psychology*: Psychology Press.
- Watt, M. (1982). What is Logo? *Creative Computing, 8*(10), 112-129.
- Weintrop, D., & Wilensky, U. (2015). To block or not to block, that is the question: Students' perceptions of blocks-based programming. *Proceedings of the 14th International Conference on Interaction Design and Children* (pp 199-208). Medford, MA, USA: ACM. doi: 10.1145/2771839.2771860
- Wilensky, U., & Reisman, K. (2006). Thinking like a wolf, a sheep, or a firefly: Learning biology through constructing and testing computational theories—An embodied modeling approach. *Cognition and Instruction, 24*(2), 171-209.
- Wilson, A., Hainey, T., & Connolly, T. (2012). Evaluation of computer games developed by primary school children to gauge understanding of programming concepts. *Proceedings of the 6th European Conference on Games-based Learning (ECGBL)*. Cork, Ireland. Retrieved from http://scratched.gse.harvard.edu/sites/default/files/evaluation_of_computer_games_developed_by_primary_school_children_to_gauge_understanding_of_programming_concepts.pdf
- Wolz, U., Hallberg, C., & Taylor, B. (2011, March). *Scrape: A tool for visualizing the code of Scratch programs*. Poster session presented at the 42nd ACM Technical Symposium on Computer Science Education. Dallas, TX, USA: ACM.
- Yarnall, L., & Kafai, Y. (1996). Issues in project-based science activities: Children's constructions of ocean software games. In *Annual Meeting of the American Educational Research Association*. New York, NY, USA. (ERIC Document Reproduction Service No. ED395819)
- Zavala, L. A., Gallardo, S. C. H., & García-Ruiz, M. Á. (2013). Designing interactive activities within Scratch 2.0 for improving abilities to identify numerical sequences. *Proceedings of the 12th International Conference on Interaction Design and Children* (pp. 423-426). New York, NY, USA: ACM. doi: 10.1145/2485760.2485831
- Zimmerman, B. J., Bandura, A., & Martinez-Pons, M. (1992). Self-motivation for academic attainment: The role of self-efficacy beliefs and personal goal setting. *American Educational Research Journal, 29*(3), 663-676.

Biographies



Jesús Moreno-León joined Universidad Rey Juan Carlos in 2012, after a decade as a Computer Science teacher, to perform research and promotion of computational thinking. As part of his dissemination activities, he co-founded and runs Programamos (We Code), a non-profit organization that promotes the development of computational thinking from early ages. He collaborates with the European Commission as Spanish ambassador for EU Code Week.



Gregorio Robles is an Associate Professor at the Universidad Rey Juan Carlos, Madrid. He earned his PhD in 2006 with a thesis on empirical software engineering research on free/libre/open source software. In addition to this line of research, he is active in the field of technology enhanced learning, in particular, in the study and evaluation of computational thinking.



Marcos Román-González, PhD in Education, is assistant professor in the National University of Distance Education (UNED, Spain). His main research lines are related to computational thinking, code-literacy, and computer science education. He is specially focused on the assessment issues of the aforementioned topics. His most recent publications include a paper in *Computers & Education* about the integration of visual programming languages across the curriculum in elementary school.