

# Visual programming languages integrated across the curriculum in elementary school: A two year case study using “Scratch” in five schools

Jose-Manuel Saez-Lopez\*, Marcos Roman-Gonzalez, Esteban Vazquez-Cano

Spanish National University of Distance Education (UNED), Spain

---

## article info

### Article history:

Received 24 May 2015

Received in revised form 4 March 2016

Accepted 5 March 2016

Available online 10 March 2016

### Keywords:

Computational thinking

Elementary education

Improving classroom teaching

Programming and programming languages

Teaching/learning strategies

## abstract

Several authors and studies highlight the benefits of the integration of Computer Science into K-12 education. Applications such as Scratch have been demonstrated to be effective in educational environments. The aim of this study is to assess the use of a Visual Programming Language using Scratch in classroom practice, analyzing the outcomes and attitudes of 107 primary school students from 5th to 6th grade in five different schools in Spain. The intervention takes place in two academic years analyzing the practice of integrating coding and visual blocks programming in sciences and arts. The “Computational concepts and computational practices” dimension details a quasi-experimental approach, which showed significant improvement regarding learning programming concepts, logic, and computational practices with an active approach. The “Learning processes and coding in primary education” dimension analyzes the practice of the experimental group through questionnaires and structured observation. In this pedagogical design, students interact and create their own content related to curricular areas with several advantages, such as motivation, fun, commitment, and enthusiasm, showing improvements related to computational thinking and computational practices. Understanding of computational concepts through an active approach, Project Based Learning, usefulness, motivation, and commitment underline the importance and effectiveness of implementing a Visual Programming Language from active methodologies in primary education. Due to the aforementioned benefits and positive results obtained in this research, it is recommended to implement a Visual Programming Language in educational settings in 5th and 6th grade in primary education through a cross-curricular implementation.

© 2016 Elsevier Ltd. All rights reserved.

---

## 1. Introduction

There is growing evidence to support the integration of Computer Science into K-12 education and students often give up Computer Science because they think it is confusing and difficult. Some teachers perceive programming to be related to training and career opportunities in technology companies, ignoring the global benefits and advantages of coding in several areas in elementary school. The purpose of the present study is to evaluate the use of Scratch in school lessons as an introduction to programming for total novices, in a younger age-group at primary school.

---

\* Corresponding author.

E-mail addresses: [jmsaezlopez@edu.uned.es](mailto:jmsaezlopez@edu.uned.es) (J.-M. Saez-Lopez), [mroman@edu.uned.es](mailto:mroman@edu.uned.es) (M. Roman-Gonzalez), [evazquez@edu.uned.es](mailto:evazquez@edu.uned.es) (E. Vazquez-Cano).

Scratch is a free programming language where you can create your own interactive stories, games, and animations. Scratch is an open-source media-rich programming environment. This program also allows students to create and develop programs related to animations, games, interfaces, and presentations that can expand understanding of computational concepts and computational practices. This visual environment enables an intuitive drag and drop method of programming which allows users to explore and create in educational settings at several levels in primary school. The aforementioned application is aimed at engaging young learners to provide an accessible starting point for learning with limited or no programming background (Good, 2011).

Today's educators are leveraging technology tools that set challenges focused on active learning in educational contexts. Project-based learning supports dynamic learning experiences leveraging the availability and use of educational technologies that can be included and integrated into the classroom. These learning approaches are focused on teaching methodology and student-centered designs.

From the perspective of the K-12 Horizon Report (Johnson, Adams Becker, Estrada, & Freeman, 2014), project-based learning, problem-based learning, inquiry-based learning, challenge-based learning, and similar methods foster more active learning experiences, both inside and outside the classroom. These approaches are student-centered, allowing learners to take control of how they engage with a subject.

Numerous studies have examined how a Flipped Classroom model has an impact on learning, with very encouraging preliminary results (Johnson et al., 2014) exploiting interactions and virtual learning environments (Saez-Lopez, Miller, Vazquez-Cano, & Domínguez-Garrido, 2015).

From the aforementioned learning models and strategies centered on the student, computational thinking entails logical analysis of data, modeling and abstractions, and implementing possible solutions. All these practices in an educational context enable students to understand how the world works and equip them with skills deemed essential in solving complex problems (Johnson et al., 2014).

Computational thinking plans and coding in education are growing in several countries due to several advantages. In 2012 there was a national initiative in Estonia oriented to integrate coding into the curriculum. United Kingdom also implemented a national mandate that children be taught computer programming in primary and secondary schools. Moreover the Finnish educational system promotes a comprehensive knowledge to be able to build technologies, to understand and create their own apps and devices. The UK Secretary of State for Education highlighted the importance of adapting the curriculum to teaching kids logical thinking to create and debug programs.

There has been a growing interest in learning to program in pedagogical contexts, driven and disseminated by organizations such as “codecademy.com” and code.org,” not only for future job opportunities and growing demands in this field, but for the educational advantages and benefits that coding in education provides.

## 2. Theoretical framework

### 2.1. Educational research principles

The present research process focused on the application of a Design Based Research strategy (Anderson & Shattuck, 2012; Dede, Ketelhut, Whitehouse, Breit, & McCloskey, 2009) that allows an intervention using complementary methods, which contribute to understanding interactions in learning processes. It is important that Design Based Research (DBR) has an influence and impact on real educative practices to justify the value of theoretical approaches.

Educational processes have been strengthened in recent decades from scientific knowledge in pedagogy across different models and methodological approaches in educational research. DBR is being utilized increasingly in educational contexts (Anderson & Shattuck, 2012, p. 24). DBR offers a “best practice” stance that has proved useful in complex learning environments, where formative evaluation plays a significant role (Dede et al., 2009, p. 6).

DBR is proposed as a strategy to innovate in educational contexts, and allows for a systematic strategy focused on learning. It is a naturalistic approach to understanding the processes of learning through informed exploration, enactment, evaluation within a local context, and development of design principles (Anderson, 2005). This approach improves the impact of educational interventions. Multiple methods and interactions constitute basic elements in this approach (Maxcy, 2003).

### 2.2. Pedagogical design and theoretical foundations: classic approaches, Project Based Learning and Bloom's taxonomy

The applied pedagogical model is based on the pedagogical implementation of cross-curricular approaches through the contemporary learning and inter-disciplinary approaches. The application of principles of instruction (Gagne, Briggs, & Wager, 1992) is elemental in this process in order to gain the attention of the students, inform them of the objectives, stimulate recall of prior learning, present the content, provide learning guidance, elicit performance (practice), provide feedback, assess performance, enhance retention, and transfer skills to the job.

Significant learning and prior learning is important from the perspective of other classic authors taken into account in this pedagogical design and collaborative learning through critical thinking (Ausubel, 1978). Social interactions in learning environments from the perspective of socio-cultural and constructivism theories (Vygotsky, 1978) are essential. Other important elements are related to interactions in social and cultural contexts in educational activities, situated learning, and active participation in learning communities and groups with an intercultural component, in this case.

Moreover, Project Based Learning (PBL) is a constructivist approach in educational activities aimed at solving problems in real contexts with opportunities for inquiry-based learning (discovery) in order to ensure that learning occurs when the subject investigates, discovers, and solves problems actively.

Learning by doing motivates students and allows them to develop oriented problem solving strategies. The approaches to learning by doing and learning by creating imply an approach to teaching and learning in gaining knowledge and skills focused on educational process. By working on self-directed projects where students think critically and communicate effectively, students are mastering core academic content aligned with 21st century skills while tackling real issues in their community and beyond (Johnson et al., 2014; Saez-Lopez & Ruiz, 2012).

In order to analyze learning processes we need categories, instruments, and taxonomies, and the degree of increase in knowledge levels serves as a key indicator for knowledge increase in general (Bloom, 1956). To assess a learning process, the degree of increase in knowledge levels must be measured.

The learning objective is to obtain high levels of knowledge, but knowledge is not simply an accumulation of knowledge for the student to remember. Since the contribution of Bloom's taxonomy, it is assumed that learning to higher levels depends on the acquisition of knowledge and skills of certain lower levels. Learning essentially involves different hierarchical levels of knowledge detailing intellectual skills into six categories:

The first and most basic level is "knowledge," which shows the memory of previously learned material through evocable facts, terms, basic concepts, and answers. The second level of knowledge must be "comprehended," which implies that students understand or interpret data based on prior knowledge. The third level of knowledge is "application," which implies that students select, transfer, and use data and principles to complete a problem or task. The fourth level is "analysis," which entails that students distinguish, classify, or relate assumptions, hypotheses, or evidence of an issue to understand the organizational structure. The following level is "synthesis," in which students integrate and combine ideas compiling information in different ways by combining elements into a new pattern and structure. The sixth and final level of knowledge is "evaluation," which deals with presentation of reviews assessing the information, validity of ideas, or quality in relation to a set of criteria.

### 23. Computational thinking and programming in educational contexts: Scratch

Wing (2006) is one of the first authors who defined the term computational thinking as "solving problems, designing systems and understanding of human behavior, using the fundamental concepts of computer science" (Wing, 2006, p. 33). The International Society for Technology in Education and the Computer Science Teachers Association (2011) define computational thinking as:

Formulating problems in a way that enables us to use a computer and other tools to help solve them; logically organizing and analyzing data; representing data through abstractions such as models and simulations; automating solutions through algorithmic thinking (a series of ordered steps); identifying, analyzing, and implementing possible solutions with the goal of achieving the most efficient and effective combinations of steps and resources; and generalizing and transferring this problem solving process to a wide variety of problems.

Computational thinking is based on processes, either by a human or a machine. The methods and computational models allow solving problems and designing systems that we could not do alone. Therefore it comes to using a computer to solve a series of problems through problem representation, prediction, and abstraction (Kafai & Burke, 2014; Sengupta, Kinnebrew, Basu, Biswas, & Clark, 2013).

Programming is not only a fundamental skill of computational science and a key tool for supporting the cognitive tasks involved in computational thinking but a demonstration of computational competencies as well (Grover & Pea, 2013), improving learners' higher-order thinking skills, and the development of algorithmic problem-solving skills (Fessakis, Gouli, & Mavroudi, 2013; Kafai & Burke, 2014).

The ability to be a creator rather than just a consumer of technology is increasingly seen as an essential skill in order to participate fully in a digital society. There is research related to computing in education (Clark, Rogers, Spradling, & Pais, 2013; Gardner & Feng, 2010; Lambert & Guiffre, 2009). But the importance of teaching computational thinking skills from an early age is a key element that has captured the attention of some researchers (Fletcher & Lu, 2009).

Computer programs tell the computer precisely what to do, step-by-step. Writing computer programs doesn't require special expertise, just clear and careful thinking. Therefore, there may be a great advantage in integrating these practices into pedagogical activities to enhance logic, math, Project Based Learning, problem solving, and critical thinking skills. There are strong reasons for students to learn to program.

Only a few studies have focused on computing in elementary school settings. The majority have focused on computer science at the high school level with a direct focus on the career path (Maya, Pearson, Tapia, Wherfel, & Reese, 2015).

Even though there is growing evidence to support the integration of Computer Science into K-12 education, there are also misconceptions and inaccurate perceptions (Armoni, 2011). Some students often give up computer science because they think it is boring, confusing, and too difficult to master (Wilson & Moffat, 2010). Visual programming solves many problems, allowing coding in primary school. Visual programming with blocks (such as Scratch) can fit only in ways that make sense, because of their shapes, so it is not possible to get error messages from the compiler. This is a great relief for introductory programming, and saves the learner from much of the headache traditionally forced on them by textual languages (Wilson & Moffat, 2010, p. 70).

Some teachers perceive programming to be related to training and career opportunities in technology companies, ignoring the global benefits and advantages in several areas. Some teachers believe that the only computing experiences are those related to programming languages such as Java or C, but they may never consider introducing computing in the earlier grades (Maya et al., 2015).

Scratch is a programming language created by the Lifelong Kindergarten group at the MIT Media Lab. The Scratch programming language offers more than 100 programming blocks, grouped into eight different categories (motion, looks, sound, pen, control, sensing, operators, and variables). This programming environment enables young people to create their own interactive stories, games, and simulations, and then share these creations in an online community with other young programmers from around the world. Pupils can program and share interactive media such as stories, games, and animation. Children learn to think creatively and collaboratively using Scratch. Coding in this interface is easier than traditional programming languages due to kids playing and interacting with the colorful blocks to create scripts.

Scratch is based on the ideas of the constructivist learning and “logo” project (Papert, 1980). This versatile application can be used to create projects containing media scripts. Images and sounds can be imported or created in Scratch using a built-in paint tool and sound recorder (Maloney, Resnick, Rusk, Silverman, & Eastmond, 2010).

Resnick, Maloney, Hernandez, Rusk, Eastmond, Brennan et al. (2009) highlight that programming is an extension of writing. The ability to program allows people to “write” new types of things, such as interactive stories, games, animations, and simulations.

Sengupta et al. (2013) developed a framework for aligning concepts of computational thinking with scientific inquiry to showcase how and why computing should be integrated into science and math instruction. They demonstrate the effectiveness of aligning these concepts in a middle school science classroom.

In the process of learning how to program, students learn mathematics (Sengupta et al., 2013) and computational concepts such as variables, loops, or conditionals. These practices can be applied transversely in primary education through project-based learning, in many subjects: math, language, arts education, science, and social science.

Brennan (2012) notes key activities that should be included in the design of learning environments mainly related to constructionism. Constructionism is grounded in the belief that the most effective learning experiences are related to active construction, socially meaningful elements, interactions with others (Papert, 1980), and elements that support thinking about one's own thinking (Kolodner et al., 2003; Papert, 1980).

Teachers and students have the perception that programming is very complicated due to the high level of abstraction of the concepts in order to program. The creators of Scratch (Resnick et al., 2009) believe that it is possible to encompass different types of projects in different contexts through a fun, meaningful, and social programming language. Papert (1980) argued that programming languages should have a “low floor” (easy to get started) and a “high ceiling” (complex projects). The Scratch programming environment and language work together to create a system that is exceptionally quick to learn (users can be programming within fifteen minutes) yet with enough depth and variety to keep users engaged for years (Maloney et al., 2010, p. 14). The children spent more time working on Scratch than with any other package they had available to them. It seems clear that Scratch succeeds very well in fostering creativity and in social sharing of the multimedia products (Weintrop et al., 2015; Wilson & Moffat, 2010).

The learner has to be able to put concepts to use in their projects and understand other students' work. Assessments should explore these multiple ways of knowing. “The intersection of computational thinking concepts and computational thinking practices leads to multiple ways of knowing” (Brennan & Resnick, 2012, p. 23).

Through Scratch, it is intended that students will be able to use programming concepts through a visual and intuitive language, because the management is performed by placing blocks of different colors and commands, which result in a product. “The Scratch programming system strives to help users build intuitions about computer programming as they create projects that engage their interests” (Maloney et al., 2010, p. 14).

Brennan and Resnick (2012) described some basic computational concepts that designers tend to use when they program:

- Sequence: To create a program in Scratch, you need to think systematically about the order of steps.
- Iteration (looping): Forever and repeat can be used for iteration (repeating a series of instructions).
- Conditional statements: If and if-else check for a condition.
- Threads (parallel execution): Launching two stacks at the same time creates two independent threads that execute in parallel.
- Event handling: For example, when key is pressed and when sprite is clicked, there are actions.
- User interface design: For example, using clickable sprites to create buttons.
- Keyboard input: Some blocks prompt users to type.

Several research studies found positive outcomes related to both attitudes about computing and computer science (Lambert & Guiffre, 2009; Lin, Yen, Yang, & Chen, 2005) and increased skills related to computational concepts (Baytak & Land, 2011; Kwon, Kim, Shim, & Lee, 2012). Maya et al. (2015) argue that research on teaching practices indicated that teachers who were initially skeptical of implementing computing found computer programs such as Scratch and Etoys to be both valuable (Clark et al., 2013) and accessible (Lee, 2011).

### 3. Aims

The main objective of the study is to analyze the benefits and possibilities of coding with a Visual Programming Language through projects and activities in primary education. The specific objectives are:

- To assess the attitudes of primary school students regarding programming, projects and content creation using a Visual Programming Language.
- To analyze creation of multimedia content, digital competence, and learning processes through Learning Centered Design and coding.
- To check students' motivation, encouragement, perceived usefulness, and results through Project Based Learning and programming.
- To analyze acquisition of basic computational programming concepts in primary education.

### 4. Method

The research process focused on an intervention using complementary methods, implementing a “Design Based Research” strategy (Anderson & Shattuck, 2012; Dede et al., 2009). The aforementioned DBR involves multiple collaborative iterations (Anderson & Shattuck, 2012). In this significant intervention, being situated in a real educational context DBR provides a sense of validity to the research and ensures that the results can be effectively used to improve practice.

The DBR approach is implemented in this particular setting consistently with this methodology. This case study is approached from mixed research methods (described in different dimensions) and it involves multiple iterations, collaborative partnership between researchers and teachers, and practical impact on the educational practice. This methodology incorporates both evaluation and empirical analyses.

From the detailed Design Based Research frame, the research applies mixed and complementary methods from quantitative and qualitative data and instruments (Fig. 1). In dimension 1 a quasi-experimental design is applied, analyzing data through *t*-test.

Dimension 2 is centered on two academic years of pedagogical practice in primary education. So it is a case study with a survey and structure observation as techniques. In this context we use classic Bloom taxonomy (1956) to assess the practice

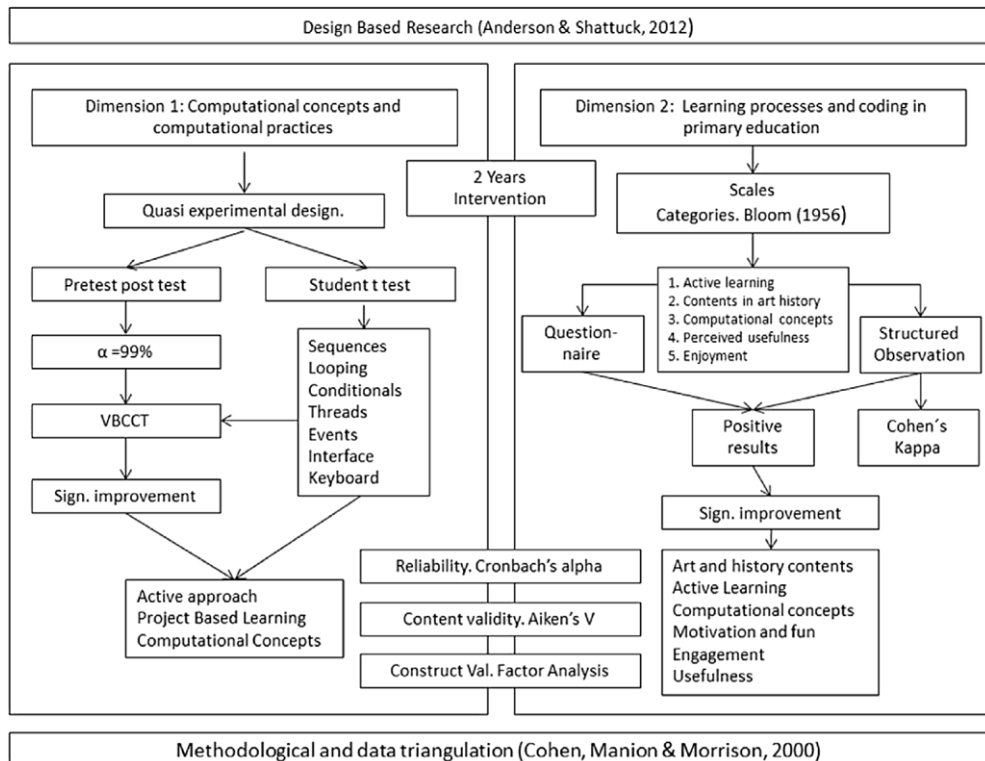


Fig. 1. Research design. Dimensions, techniques, and instruments.

and a Bloom Rubric (Fu, Wu, & Ho, 2009) to analyze learning processes and results. In this dimension the evaluation was based on a naturalistic evaluation model approach, which is conducted with the collaboration of the participant students and teachers (Guba & Lincoln, 1981). Therefore, suggested evaluation is an applied synchronized field survey, which combines qualitative and quantitative evaluation methods and structured observation.

Data triangulation ensures that there is evidence to support the validity of results and minimize error variance (Goetz & LeCompte, 1988). The data triangulation of Cohen, Manion, and Morrison (2000) was implemented using quantitative information collected in tests and scale.

Triangulation of data is developed from test results in dimension 1 and the implementation and evaluation of the scale administered to 6th grade students. To analyze the scale results, this study assumes that the degree of learning determines the development of the knowledge process from Bloom's (1956) framework and rubric.

Using information provided from taxonomies (Näsström, 2009), practice is designed to harnesses the potential to understand and create with the Scratch application, which facilitates the work with codes and programs (scripts) to create multimedia content (Brennan & Resnick, 2012; Maloney et al., 2010) with an active student-centered approach.

Independent variables are basically: grade (primary school, 6th grade) and gender. Controlled variables related to attitudes are perceived usefulness and self-efficacy. Learning is related to history and computational contents.

#### 4.1. Participants

The study sample consists of 107 primary school students from sixth grade in five different schools in the Castilla-La Mancha and Madrid regions in Spain. Regarding gender, 60.7% are girls and 39.3% are boys. Response rate is 86.991% so sample mortality is 13.008%. We also have a control group from one school with 32 primary school students, which response rate is 91.428% and sample mortality is 8.572%. Contingency analysis (Chi square) is not detailed because there are no significant differences regarding gender or school. The sample is not randomized; therefore dimension 1 is a quasi-experimental design. Dimension 2 analyzes the practice of the aforementioned experimental group through two academic years (fifth and sixth grade in primary school).

#### 4.2. Program implementation

Intervention design is a key feature of the quality and results of research projects. It is important to document creation and implementation during intervention so readers of the research can judge for themselves the possibility of achieving equivalent results from the use of interventions in their own contexts.

From an international perspective, these practices are related to K-12 curriculum standards, including the Common Core State Standards, the CSTA K-12 Computer Science Standards and ISTE NETS Common Core State Standards for Mathematics 2010. In Spain, the curriculum framework is based on primary school standards (Spanish Ministry of Education and Culture and Sports, MECD, 2014) and from important perspectives of digital competence framed in key competences for lifelong learning (European Parliament and Council, 2006).

Application takes place in 5th and 6th grade in primary education over the academic years 2013–14 and 2014–15 in 20 one-hour sessions integrated in sciences and arts with the use of a Visual Programming Language, specifically using integrated multimedia resources with the Scratch application (Fig. 2).

Competences in these sessions are: Mathematical competence, science competence, digital competence, learning to learn, and cultural competence (European Parliament and Council, 2006). Computational concepts and practices involved are the following (Table 1).

Regarding the processes of construction in computational practices in the classroom, we focus on “experimenting and iterating,” which refers to students developing projects step by step, trying new contents and elements, applying different concepts, and revising them.

#### 4.3. Instruments and reliability

Through the DBR approach, we apply mixed methods research using a variety of tools and techniques in the intervention consistent with the research design. “It is perfectly logical for researchers to select and use differing methods, selecting them as they see the need, applying their findings to a reality that is both plural and unknown” (Maxcy, 2003, p. 59).

The intervention comprises two academic years (2013–14, 2014–15) with programmed activities. Students created their own material based on the project goals applying methodological strategies described in the theoretical framework. Multimedia, coding, and communications enabled several possibilities analyzed in this research. The present study proposes two dimensions that address the research objectives.

The first dimension measures Visual Blocks Creative Computing Test (VBCCT) results through a quasi-experimental method. Qualitative validity of content provided by nine expert judges provides a value of Aiken V ( $\sum S/[n(n-1)])$  greater than 0.8 in all items. Pretest/posttest design measures computational concepts and computational practices. Construct validity was examined by exploratory factor analysis, using the criterion of extraction of eigenvalues  $>1$ , and the method of varimax rotation. Moreover, an 8.36 value of Cronbach reliability is acceptable (Hair, Anderson, Tatham, & Black, 1998). The aforementioned test has 40 items with a structured and progressive sequence. It assesses the understanding of

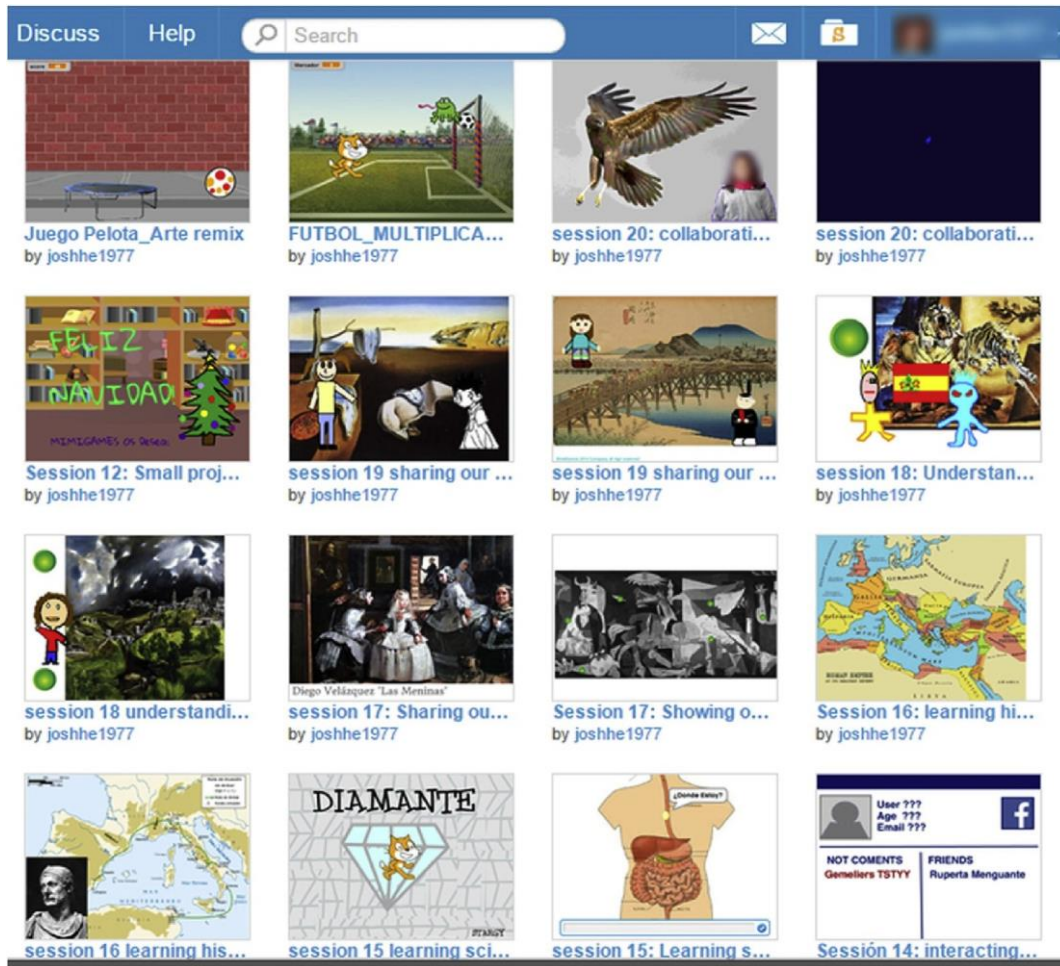


Fig. 2. Creative computing activities from 2013 to 2015. Source: <http://edresearch.hypotheses.org/90>.

computational concepts, the use of different commands and visual blocks that allow the understanding of computational practices to apply computer animations, games, and creations in educational settings. This test covers aspects related to computational concepts and computational practices. Students answer items related to sequences, loops, conditional statements, parallel execution, coordination, event handling, and keyboard input. Experimenting and iterating is important to understand and answer the test's items with success.

The second dimension analyzes a questionnaire that analyzes learning processes and students' attitudes. This kind of research is intended to describe the individual experience in particular environments (Creswell, 2003). Students filled out this questionnaire pertaining to their perception of the designed scales (Table 5). Another instrument used for this dimension has been structured-systematic observation. A coding method based on the defined categories (Table 5) and Bloom's taxonomy (1956) was used and it has already been explained in various sections of this paper (see Table 2). The use of structured-systematic observation as a research method has the advantage of accessing directly what students have experienced while they were carrying out the activities. Thus, the application of this data collection method throughout the whole educational intervention ensures that data is collected without researchers' direct interaction with participants.

During the conduct of structured-systematic observation, classifying subjects or objects into predefined classes or categories is a rather common activity. The reliability of this classification process can be established by asking two individuals referred to as raters, to independently perform this classification with the same set of subjects or objects. By accomplishing this task, these two individuals will have just participated in what is called an inter-rater reliability experiment expected to produce two categorizations of the same subjects or objects. The extent to which these two categorizations coincide represents what is often referred to as inter-rater reliability (Gwet, 2014).

Inter-rater reliability or concordance is the degree of agreement among raters. The statistic used for this measurement is Cohen's kappa with the teacher (in each school) and the researcher as raters, taking into account the amount of agreement that could be expected. Cohen's kappa is one of the most commonly used statistics to test inter-rater reliability, like most

Table 1  
Computational concepts and practices.

Computational concepts	Computational practices
Sequence	Experimentation and iteration
Iteration (Looping)	
Conditional Statements	
Threads (Parallel Execution)	
Event Handling	
User Interface Design	
Keyboard Input	

Source: Brennan, Balch, & Chung, 2014.

correlation statistics, the kappa can range from  $-1$  to  $1$ . We get a value regarding the relative observed agreement among raters. Only items with Cohen's kappa values over  $0.60$  are accepted in this research. All items under this value were eliminated. Cohen (1960) suggested: values  $0$  as indicating no agreement and  $0.01$ – $0.20$  as none to slight,  $0.21$ – $0.40$  as fair,  $0.41$ – $0.60$  as moderate,  $0.61$ – $0.80$  as substantial, and  $0.81$ – $1.00$  as almost perfect agreement (McHugh, 2012).

Methodological and data triangulation consistent with a DBR design provides data from different sources, techniques, and instruments in order to increase validity. It is recommended to use more than one method to enhance the validation process. Inter-method and multi-method and independent measures reach the same conclusions, providing validity and reliability.

The survey questionnaire includes five scales. All participants completed the aforementioned paper questionnaire after the experiment in the classroom. Content validity provided by nine expert judges gives a value of Aiken  $V$  higher than  $0.7$  in all items. Therefore, relevance and appropriateness of the instrument in qualitative validation is acceptable. Moreover, construct validity was examined by exploratory factor analysis, using the criterion of extraction of eigenvalues  $>1$ , and the method of varimax rotation. Regarding value of Cronbach reliability,  $7.89$  is acceptable.

- The scale active learning 1 contains five questions, present in Hiltz, Coppola, Rotter, and Turoff (2000).
- Scales 2, “Contents in art history,” and 3, “Computational concepts and Gamification,” are based on a study of Saez-Lopez et al., (2015).
- Scale 4 for perceived usefulness consists of three questions adapted from Davis, Bagozzi, and Warshaw (2002).
- Scale 5 for “fun” during the learning activities consists of five questions adapted from the scale created by Laros and Steenkamp (2005).

With this intervention, Scratch is integrated across two academic years in the Spanish education system in social sciences and art education. Bloom's taxonomy is considered when assessing the first subscale criteria (Table 2). The degree of increased levels of knowledge serves as a key indicator for increased knowledge in general (Bloom, 1956). To evaluate the learning process, the degree of increase in knowledge levels should be measured. Learning essentially involves different levels of knowledge. Bloom's taxonomy addresses the acquisition of knowledge as opposed to information based on memorization of the student.

Bloom (1956) highlighted a hierarchy of knowledge in six categories which are organized into levels of difficulty. The first and most basic level is “knowledge,” which is the data or information that one can remember. The second category of knowledge should be “understood,” implying that students understand the meaning, translation, interpolation, and interpretation of instructions. The third level of knowledge is “application,” which involves the use of a concept in a new situation. The fourth level is the “analysis” that focuses on separation of concepts in their parts in order to understand the organizational structure. Along with the analytical skills, students must be able to participate in “synthesis” by building a structure or pattern from diverse elements. The sixth and final level of knowledge is “evaluation,” which deals with the ability to make judgments about the value of ideas or materials.

## 5. Results

### 5.1. Dimension 1: computational concepts and computational practices

Dimension 1 is centered on the application of a quasi-experimental design, analyzing data through a student  $t$ -test with a paired samples test. We have applied Visual Blocks Creative Computing Test (VBCCT) with a pretest and posttest design, which allowed us to ascertain whether there were significant improvements before and after the implementation of the project. Normality is assumed due to sample size and the Kolmogorov-Smirnov test. The significance level ( $\alpha$ ) is  $0.01$ .

From the results of the student  $t$ -test administered, it can be stated that there are significant improvements in the results of the aforementioned test, so the implemented program improves the ability of students to understand programming concepts and logic and create multimedia products related to curriculum contents.

Pre- $t$ -test values are obtained from conventional practices before integrating coding in educational contexts (mean is  $7.95$ ). Moreover, values obtained in the post- $t$ -test give data obtained after the application of the intervention (mean is  $30.07$ ),



Table 2  
Rubric based on Bloom's taxonomy (Bloom, 1956).

Grade				
Failed	Passed	Acceptable	Good	Excellent
The question did not reflect any deliberation on the course material. The question makes no sense. (no learning)	The student relied completely on course material with no creative thinking of his/her own. (memorization)	The student set up a question and provided a rational, logical answer. (comprehension)	The student understood certain aspects of the course material to a degree that he/she created a question that required deductive thinking. (application/analysis)	The student synthesized course content and his/her pre-existing knowledge to generate a creative question that provided added value to the course content. (synthesis/evaluation)

Source: Fu et al., 2009, p. 561.

Table 3  
Paired differences. Student *t*-test.

	Mean	Std. Deviation	Std. Error mean	Upper α ¼ 99%	Lower α ¼ 99%	t	df	Sig
POSTT_(VBCCT) – PRETT_(VBCCT)	–22.121	5.399	0.522	–23.490	–20.753	–42.386	106	0.000

emphasizing statistically significant differences ( $p < 0.001$ ) at a significance level of 99% between paired samples (Table 3). The values in the Visual Blocks Creative Computing Test (VBCCT) stressed the importance of an educational design that includes Visual Programming Language to understand the elements of logic, mathematics, and content creation, providing improvements as highlighted in this analysis.

Moreover, we compare the independent post-test independent samples, the experimental group (mean is 30.07), and the control group (mean is 10.28). We apply Levene's Test for equality of variances (Table 4). We obtain a value of 0.008 so equal variances are assumed at a significance level ( $\alpha/0.01$ ). There are statistically significant differences ( $p < 0.001$ ) at a significance level of 99% between control group and experimental group.

## 5.2 Dimension 2: learning processes and coding in primary education

This dimension highlights the case study which analyzes recollected data from the mentioned scales and structured observation through two academic years in primary education (Table 5).

Classic Bloom's taxonomy (1956) is addressed as a framework to assess the practice using a Bloom Rubric (Fu et al., 2009) to analyze learning processes and results. In this dimension the evaluation was based on a naturalistic evaluation model approach, which is conducted with the collaboration of the participant students and teachers (Guba & Lincoln, 1981). Therefore, the suggested evaluation is an applied synchronized field survey, which combines qualitative and quantitative evaluation methods and structured observation.

Data triangulation enables validity of results and minimizes error variance (Cohen et al., 2000; Goetz & LeCompte, 1988). Different techniques and instruments highlighted similar results in several categories.

Results obtained in the scale 1 "active learning" from the questionnaire and structured observation show values higher than 4 related to the interest of the subject and students' active participation in these processes (1.3 and 1.4). Concerning learning of factual materials and the ability to communicate clearly, the values are somewhat lower than 3.5 (1.1 and 1.2). These are positive values but they are somewhat lower than the factors mentioned above.

The questionnaire and the observation technique disagree to some extent in item 1.5 which emphasizes assignments that aided the student's learning. Although both values are positive, they are considerably higher from the perspective of the students' questionnaire.

In short, in scale 1, which refers to active learning, we got values close to 4, so the obtained values in this scale are considerably high.

Regarding art and history contents, in scale 2, values are over 3 concerning understanding of artistic elements in paintings, biographical and historical contents, cultural and artistic competence, and the ability to understand artistic expressions. Therefore depending on the category, students are capable of understanding the concepts (Bloom, 1956). The analysis of historical and artistic content gets higher values when integrating to multimedia content. Regarding contents related to art and history mean is around 3.65 from questionnaire and observation data.

Regarding computational concepts, there are values higher than 4 related to working with sequences, loops, parallelism, events, sharing, and fun when creating these products. There is a high correlation between administered instruments. Concerning communication (item 1.5) with created content, the value is somewhat below 4, and even lower when analyzed by observation technique (3.41).

For perceived usefulness, detailed in scale 4, there are values above 4 in all items, so the efficiency, utility, and improvement in the learning process are clearly reflected and detailed with these values with both instruments.

Table 4  
Independent samples. Experimental group and control group. Student *t*-test.

	Levene's test for equality of Variances		<i>t</i> -test for equality of means				
	F	Sig.	t	df	Sig. (2-Tailed)	Mean difference	Std. Error difference
POSIT_(VBCCT) Equal variances assumed	7.238	0.008	23.433	137	0.000	19.794	0.845

Table 5  
Dimension 2 scales. Values in questionnaire and structured observation.

Scales	Items	Questionnaire		Structured observation	
		M	$\bar{x}$	M	$\bar{x}$
1. Active learning	1. Learned many factual materials	3.34	3.93	3.46	3.72
	2. Improved ability to communicate clearly	3.38		3.43	
	3. Became more interested in the subject	4.20		4.11	
	4. Participated actively	4.45		4.07	
	5. Assignments aided the student's learning	4.31		3.54	
2. Contents in art history	1. Understood artistic elements in paintings	3.60	3.69	3.54	3.45
	2. Learned biographical and historical contents of Spanish painters	3.70		3.39	
	3. Increased cultural and artistic competence to understand paintings	3.52		3.50	
	4. Improved the ability to understand artistic expressions from different eras	3.50		3.41	
	5. Analyzed historical and artistic content in paintings	4.14		4.41	
3. Computational concepts	1. Understood sequences with combined characters, backgrounds, and elements	4.67	4.42	4.48	4.27
	2. Included loops in programming to allow a proper multimedia product	4.64		4.46	
	3. Added parallelism and events that allow the creation of interface	4.37		4.35	
	4. Improved ability to share and play with the content created	4.31		4.46	
	5. Acquired the ability to communicate and express through the content created	3.93		3.41	
	6. Increased fun to learn art history with games and animations	4.61		4.46	
4. Perceived usefulness	1. The courseware increased the efficiency of my learning process	4.06	4.55	4.72	4.63
	2. The courseware helped improve my learning performance	4.96		4.52	
	3. The courseware was useful	4.65		4.67	
5. Enjoyment	1. I was happy	4.74	4.64	4.80	4.81
	2. I enjoyed the activity	4.65		4.87	
	3. I was enthusiastic	4.60		4.80	
	5. I felt motivated	4.79		4.76	
	6. I was relaxed and comfortable	4.53		4.85	

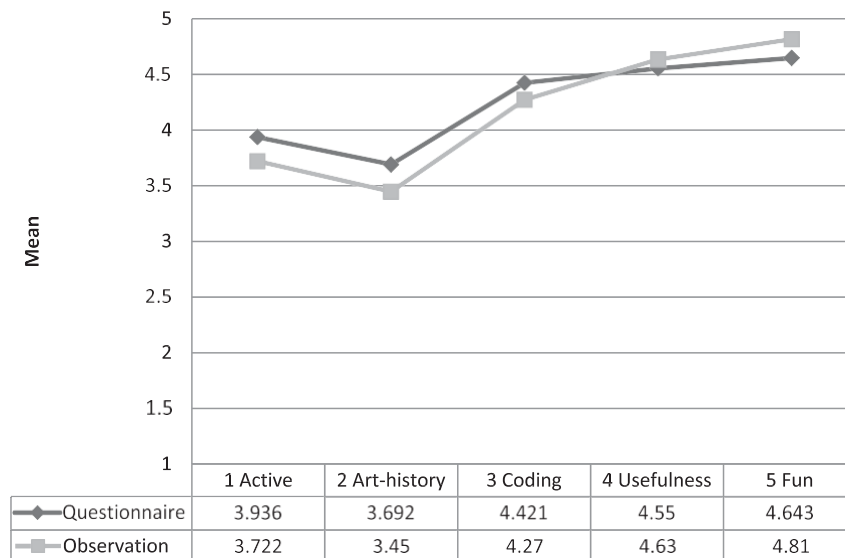


Fig. 3. Values in scales. Questionnaire and structured observation.

Scale 5 named fun obtained very high values, higher than 4 and almost 5 in all items. Therefore the students are enthusiastic, motivated, relaxed, and happy to work with the approach proposed in the intervention.

Similar results are obtained from the application of the questionnaire and structured observation. Values of all scales are positive, with particularly high values in regard to fun, utility, and computational concepts (Fig. 3).

## 6. Conclusions

In a case study with 107 primary school students working in art and social sciences with a Visual Programming Language (Scratch), we have observed benefits of using programming in an educational context besides the programming knowledge. We detail the implementation of creative computing as an intervention over two years, describing evidence and results from different instruments. There are positive values obtained mainly from factors such as active learning, art and history contents, computational concepts, useful, and fun. From data analysis this research concludes:

1. An active pedagogical approach using a Visual Programming Language significantly improves several elements: learning programming concepts, logic, and computational practices (Dimension 1, student *t*-test, VBCCT).
2. Students and observers point out that working with visual programming through projects provides fun, motivation, enthusiasm, and commitment from the student (scale 5).
3. Perceived usefulness in these practices (scale 4) and the computational concepts addressed (scale 3) obtained considerably higher results with average values greater than 4.5. Therefore, the descriptive analysis makes clear the utility and the possibility of learning sequences, loops, parallelism, and events, and sharing content (items 3.1, 3.2, 3.3, and 3.4). Fun (item 3.6) and communicative possibilities (item 3.5) are present in this educational process.
4. Project Based Learning carried out in the intervention enables an active approach (scale 1). This factor obtained values around 4, so active learning is essential and important in this process. The active approach gets positive results and stands out through a teaching methodology centered on the student to develop projects and creations.
5. With regard to art and history contents (scale 2), results are a little bit higher than 3; therefore, according to the proposed category, students achieve an understanding and comprehension of the aforementioned concepts that is consistent with Bloom's classical taxonomy (1956).

The values obtained in the study show statistically significant improvements in the understanding of computational concepts and computational practices in this educational stage, which suggests recommending to the educational authorities to implement programming in educational settings in 5th and 6th grade in primary education. Students are prepared to address computational practices and create their own content related to curricular areas (in response to the second research aim of this study). Consistent with the intervention in this research a cross-implementation in all areas is recommended, especially in social sciences and arts, given the characteristics of the visual content presented by these areas which enables creating colorful, dynamic, and motivating projects from an active perspective. These insights answer the research aim related to analyzing the acquisition of basic computational programming concepts in primary education.

Motivation, fun, commitment, and enthusiasm of the students through this pedagogical approach are reflected in the analysis of results. Students are totally in favor of this pedagogical design, highlighting usefulness and active learning provided by this approach. Therefore, students' attitudes and motivation (the first and third aim of this study) are considerably high and positive in these contexts when integrating projects and visual programming.

In short, the importance of an educational design that includes Visual Programming Language programming to understand the elements of logic, mathematics, and content creation in art and history, brings significant improvements, as this research demonstrates. Understanding of computational concepts, active approach, Project Based Learning, usefulness, motivation, and commitment underline the importance and effectiveness of implementing visual programming from active methodologies in primary education.

## Author note

Dr Jose Manuel Saez Lopez.

ORCID: [orcid.org/0000-0001-5938-1547](https://orcid.org/0000-0001-5938-1547) Dr Jose Manuel Saez Lopez is professor at the National University of Distance Education (UNED), Spain. He holds a doctorate in Educational Models and ICT from the UNED. His scientific and academic work has been published in 36 peer-reviewed journals (four JCR and eight Scopus). His research lines are the integration of educational technology in learning processes, analyzing virtual learning environments, the educational use of interactive video conferencing, methodological strategies, Game Based Learning, Coding in elementary school and the academic use of Open Access. He is accredited as a "contracted doctor lecturer" by ANECA, the National Agency for Quality Assessment and Accreditation of Spain. He has been recognized as a Microsoft Expert Educator 2014 and Microsoft Innovative Expert 2015.

Partner in "Escuela 2.0, computer per child" project from 2011 to 2013. Partner in European research Project "LLL LOGIC".

Book chapters (7) related to teaching methods and integration of educational technology. Dissemination in conferences: INTED 2011 (Valencia), ELSE conference 2011 (Bucarest), Congresos de Pizarra digital (II, III y IV. Madrid), Congreso de competencias basicas 2011 (Ciudad Real), II Congreso de innovacion en educacion primaria (Univ Murcia), Scratch Conference

(Barcelona) 2013, XV congreso de pedagogía, 2012 (Burgos), Congreso I. Videojuegos, 2013 (Caceres), congreso InterESTRATIC 2013, 2014, 2015 (Madrid).

## References

- Anderson, T. (2005). Design-based research and its application to a call centre innovation in distance education. *Canadian Journal of Learning and Technology*, 31(2), 69–83. Retrieved from [http://auspace.athabascau.ca:8080/dspace/bitstream/2149/741/1/design\\_based\\_research.pdf](http://auspace.athabascau.ca:8080/dspace/bitstream/2149/741/1/design_based_research.pdf).
- Anderson, T., & Shattuck, J. (2012). Design-based research: a decade of progress in education research? *Educational Researcher*, 41, 16–25.
- Armoni, M. (2011). The nature of CS in K-12 curricula: the roots of confusion. *ACM Inroads*, 2(4), 19–20. <http://dx.doi.org/10.1145/2038876.2038883>.
- Ausubel, D. (1978). In defense of advance organizers: a reply to the critics. *Review of Educational Research*, 48, 251–257.
- Baytak, A., & Land, S. M. (2011). An investigation of the artifacts and process of constructing computer games about environmental science in a fifth grade classroom. *Educational Technology Research and Development*, 59, 765–782. <http://dx.doi.org/10.1007/s11423-010-9184-z>.
- Bloom, B. S. (1956). *Taxonomy of educational objectives: The classification of educational goals, handbook 1 cognitive domain*. New York: David McKay Co. Inc.
- Brennan, K. (2012). *Best of both worlds: Issues of structure and agency in computational creation, in and out of school* [Doctoral dissertation]. MIT. Retrieved from [http://web.media.mit.edu/~kbrennan/files/dissertation/Brennan\\_Dissertation.pdf](http://web.media.mit.edu/~kbrennan/files/dissertation/Brennan_Dissertation.pdf).
- Brennan, K., Balch, C., & Chung, M. (2014). *Creative computing*. Retrieved from <http://scratched.gse.harvard.edu/guide/>.
- Brennan, K., & Resnick, M. (2012). *Using artifact-based interviews to study the development of computational thinking in interactive media design*. American Educational Research Association Meeting. Vancouver, BC: Canada. Retrieved from [http://web.media.mit.edu/~kbrennan/files/Brennan\\_Resnick\\_AERA2012\\_CT.pdf](http://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf).
- Clark, J., Rogers, M. P., Spradling, C., & Pais, J. (2013). What, no canoes? Lessons learned while hosting a Scratch summer camp. *Journal of Computing Sciences in Colleges*, 28, 204–210.
- Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1), 37–46.
- Cohen, L., Manion, L., & Morrison, K. (2000). *Research methods in education*. London and New York, NY: Routledge Falmer.
- Creswell, J. W. (2003). *Research design: qualitative, quantitative, and mixed methods approach*. Thousand Oaks, CA: Sage.
- Davis, F. D., Bagozzi, R. P., & Warshaw, P. R. (2002). User acceptance of computer technology: a comparison of two theoretical models. *Management Science*, 35(8), 982–1003.
- Dede, C., Ketelhut, D., Whitehouse, P., Breit, L., & McCloskey, E. (2009). A research agenda for online teacher professional development. *Journal of Teacher Education*, 60(1), 8–19.
- European Parliament and Council. (2006). *Key competences for lifelong learning – A European framework*. Official Journal of the European Union on 30 December 2006/L394. Retrieved from [http://eur-lex.europa.eu/LexUriServ/site/en/oj/2006/l\\_394/l\\_39420061230en00100018.pdf](http://eur-lex.europa.eu/LexUriServ/site/en/oj/2006/l_394/l_39420061230en00100018.pdf).
- Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5–6 year old kindergarten children in a computer programming environment: a case study. *Computers & Education*, 63, 87–97. Retrieved from <http://dx.doi.org/10.1016/j.compedu.2012.11.016>.
- Fletcher, G., & Lu, J. (2009). Human computing skills: rethinking the K-12 experience. *Communications of the ACM – Association for Computing Machinery – CACM*, 52(2), 23–25.
- Fu, F.-L., Wu, Y.-L., & Ho, H.-C. (2009). An investigation of cooperative pedagogic design for knowledge creation in web-based learning. *Computers & Education*, 53(3), 550–562.
- Gagne, R. M., Briggs, L. J., & Wager, W. W. (1992). *Principles of instructional design* (4th ed.). Fort Worth, TX: Harcourt Brace Jovanovich.
- Gardner, M. K., & Feng, W. C. (2010). Broadening accessibility to computer science for K-12 education. In *Proceedings of the 15th annual conference on innovation and technology in computer science education* (pp. 229–233). ACM. <http://dx.doi.org/10.1145/1822090.1822155>.
- Goetz, J. P., & LeCompte, M. D. (1988). *Ethnography and qualitative design in educational research*. Madrid: Ediciones Morata.
- Good, J. (2011). Learners at the wheel: novice programming environments come of age. *International Journal of People-Oriented Programming*, 1(1), 1–24. Retrieved from <http://dx.doi.org/10.4018/ijpop.2011010101>.
- Grover, S., & Pea, R. (2013). Computational thinking in K-12, a review of the state of the field. *Educational Researcher*, 42(1), 38–43.
- Guba, E. G., & Lincoln, Y. S. (1981). *Effective evaluation*. San Francisco: Jossey-Bass.
- Gwet, K. L. (2014). *Handbook of inter-rater reliability: The definitive guide to measuring the extent of agreement among raters*. Gaithersburg, MD: Advanced Analytics, LLC.
- Hair, J. F., Anderson, R. E., Tatham, R. L., & Black, W. C. (1998). *Multivariate data analysis* (5th ed.). Upper Saddle River, New Jersey: Prentice Hall.
- Hiltz, S. R., Coppola, N., Rother, N., & Turoff, M. (2000). Measuring the importance of collaborative learning for the effectiveness of ALN: a multi-measure, multi-method approach. *Journal of Asynchronous Learning Networks*, 4(2), 103–125.
- International Society for Technology in Education and the Computer Science Teachers Association. (2011). *Operational definition of computational thinking for K-12*. Retrieved from <http://csta.acm.org/Curriculum/sub/CurrFiles/CompThinkingFlyer.pdf>.
- Johnson, L., Adams Becker, S., Estrada, V., & Freeman, A. (2014). *NMC Horizon Report: 2014 K-12 edition*. Austin, Texas: The New Media Consortium. Retrieved from <http://www.nmc.org/pdf/2014-nmc-horizon-report-he-EN.pdf>.
- Kafai, Y. B., & Burke, Q. (2014). *Connected code: why children need to learn programming*. MIT Press.
- Kolodner, J., Camp, P., Crismond, D., Fasse, B., Gray, J., Holbrook, J., et al. (2003). Problem-based learning meets case-based reasoning in the middle-school science classroom: putting learning by design into practice. *Journal of the Learning Sciences*, 12(4), 495–547.
- Kwon, D. Y., Kim, H. S., Shim, J. K., & Lee, W. G. (2012). Algorithmic bricks: a tangible robot programming tool for elementary school students. *Education. IEEE Transactions*, 55(4), 474–479. <http://dx.doi.org/10.1109/TE.2012.2190071>.
- Lambert, L., & Guiffre, H. (2009). Computer science outreach in an elementary school. *Journal of Computing Sciences in Colleges*, 24(3), 118–124.
- Laros, F. J. M., & Steenkamp, J.-B. E. M. (2005). Emotions in consumer behavior: a hierarchical approach. *Journal of Business Research*, 58(10), 1437–1445.
- Lee, Y. J. (2011). Empowering teachers to create educational software: a constructivist approach utilizing Etoys, pair programming and cognitive apprenticeship. *Computers & Education*, 56(2), 527–538. <http://dx.doi.org/10.1016/j.compedu.2010.09.018>.
- Lin, J. M. C., Yen, L. Y., Yang, M. C., & Chen, C. F. (2005, June). Teaching computer programming in elementary schools: a pilot study. In *National educational computing conference*.
- Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The Scratch programming language and environment. *ACM Transactions on Computing Education*, 10(4), 1–15.
- Maxcy, S. J. (2003). Pragmatic threads in mixed methods research in the social sciences: the search for multiple modes of inquiry and the end of the philosophy of formalism. In A. Tashakkori, & C. Teddlie (Eds.), *Handbook of mixed methods in social and behavioral research* (pp. 51–89). Thousand Oaks, CA: Sage.
- Maya, I., Pearson, J. N., Tapia, T., Wherfel, Q. M., & Reese, G. (2015). Supporting all learners in school-wide computational thinking: a cross-case qualitative analysis. *Computers & Education*, 82, 263–279.
- McHugh, M. L. (2012). Interrater reliability: the kappa statistic. *Biochemia Medica*, 22(3), 276–282.
- Näsström, G. (2009). Interpretation of standards with Bloom's revised taxonomy: a comparison of teachers and assessment experts. *International Journal of Research & Method in Education*, 32, 39–51.
- Papert, S. (1980). *Mindstorms: children, computers, and powerful ideas*. New York, NY: Basic Books.
- Resnick, M., Maloney, J., Hernandez, A. M., Rusk, N., Eastmond, E., Brennan, K., et al. (2009). Scratch: programming for all. *Communications of the ACM*, 52(11). Retrieved from <http://web.media.mit.edu/~mres/papers/Scratch-CACM-final.pdf>.

- Saez-Lopez, J. M., Miller, J., Vazquez-Cano, E., & Domínguez-Garrido, M. C. (2015). Exploring application, attitudes and integration of video games: minecraftEdu in middle school. *Educational Technology & Society*, 18(3), 114–128. Retrieved from [http://www.ifets.info/journals/18\\_3/9.pdf](http://www.ifets.info/journals/18_3/9.pdf).
- Saez-Lopez, J. M., & Ruiz, J. M. (2012). Estrategias metodológicas, aprendizaje colaborativo y tic: un caso en la escuela complutense latinoamericana. *Revista Complutense de Educacion*, 23, 115–134. Retrieved from <http://revistas.ucm.es/index.php/rced/article/view/39105>.
- Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: a theoretical framework. *Education and Information Technologies*, 18, 351–380.
- Spanish Ministry of Education and Culture and Sports, MECD. (2014). *Royal Decree 126/2014, whereby the core curriculum of primary education is established*. Retrieved from <http://www.boe.es/boe/dias/2014/03/01/pdfs/BOE-A-2014-2222.pdf>.
- Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. Cambridge, MA: Harvard University Press (Chapter 6: Interaction between learning and development).
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., et al. (2015). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, 10. <http://dx.doi.org/10.1007/s10956-015-9581-5>.
- Wilson, A., & Moffat, D. C. (2010). Evaluating Scratch to introduce younger school children to programming. In *Proceedings of the 22nd Annual Psychology of Programming Interest Group (Universidad Carlos III de Madrid, Leganes, Spain)*.
- Wing, J. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35. <http://dx.doi.org/10.1145/1118178.1118215>.