International Journal on E-Learning

October 2017 Volume 16, Number 4

Editors

Gary H. Marks

Search this issue

Keywords, title, or author names Go

Table of Contents

Number of articles: **6**

Monica Banzato & Paolo Tosato, University Ca' Foscari, Venice - Italy, Italy

In Italy, teaching coding at primary and secondary levels is emerging as a major educational issue, particularly in light of the recent reforms now being implemented. Consequently, there has been ... More (/p/152245)

pp. 349-369

¶ View Abstract (/p/152245) | Download Full Text (/d/152245) | Add to Collection

What do instructional designers in higher education really do? (/p/150980)

Swapna Kumar & Albert Ritzhaupt, University of Florida, United States

What do instructional designers in higher education really do? With the rise in online courses and programs in higher education, this question is especially important. We interviewed eight... More (/p/150980)

pp. 371-393

¶ View Abstract (/p/150980) | Download Full Text (/d/150980) | Add to Collection

The contribution of online (web-based) courses as perceived by students with learning disabilities and “average” students (/p/151917)

Ilana Ronen & Miri Shonfeld, Kibbutzim College of Education, Technology and Arts, Israel

This quantitative, descriptive study examines the perceived contribution of online courses taken in college as evaluated by 53 “average” students and 32 students with learning disabilities (LD... More (/p/151917)

pp. 395-416

¶ View Abstract (/p/151917) | Download Full Text (/d/151917) | Add to Collection

Student preferences and concerns about supplemental instructional material in CS0/CS1/CS2 courses (/p/151732)

Giovanni Vincenti, University of Baltimore, United States; J. Scott Hilberg, Towson University, United States; James Braman, Community College of Baltimore County, United States

Abstract: The concept of flipping the classroom is slowly gaining tractions at all levels of education. An ever-growing set of resources gives unprecedented access to Information Technology (IT),... More (/p/151732)

pp. 417-441

¶ View Abstract (/p/151732) | Download Full Text (/d/151732) | Add to Collection

International Jl. on E-Learning (2017) **16**(4), 349-369

An Exploratory Study of the Impact of Self-Efficacy and Learning Engagement in Coding Learning Activities in Italian Middle School

MONICA BANZATO AND PAOLO TOSATO
Ca' Foscari University of Venice, Italy
banzato@unive.it
ptosato@unive.it

In Italy, teaching coding at primary and secondary levels is emerging as a major educational issue, particularly in light of the recent reforms now being implemented. Consequently, there has been increased research on how to introduce information technology in lower secondary schools. This paper presents an exploratory survey, carried out through an intensive workshop on coding, which was designed to introduce the basic principles of computer science to be included in the future school curriculum. Specifically, the paper examines three key aspects of the coding unit for lower secondary schools: (a) the students' perception of their self-efficacy in carrying out coding; (b) the involvement of students' cognition and perception; (c) the principal obstacles students might encounter while coding. The results are encouraging as they demonstrate that coding turns out to be both highly interesting and motivating for students.

Keywords: coding, education, middle school, self-efficacy, user engagement.

INTRODUCTION

Coding demands a high cognitive and metacognitive effort: to find solutions to problems never seen before requires processing and organization of information skills, evaluating and monitoring of strategies and, above all, capabilities of abstraction, representation and generalization. In short, it means developing critical, creative and effective thinking [1].

These are cognitive processes that require high endurance and concentration [2][3][4], for which it is necessary to train students, instilling motivation and helping them to overcome their greatest difficulties.

It is, therefore, important to develop positive and successful experiences that can motivate and excite students [5] and increase their sense of self-efficacy [6].

In fact, a high sense of self-efficacy can serve as a "protective shield" in the most difficult moments of the development of a particular task [7], helping to ward off despair and resignation when facing increasingly complex problems.

Given the specialized nature of coding, which often requires patience and perseverance in the search for a solution to a problem, but also in correcting and testing the solution itself, self-efficacy and student engagement are the crucial factors that determine success in the study of computer science (henceforth, CS).

When teaching coding, it is important to create positive experiences in order to support the sense of self-efficacy and learning engagement which depends upon interaction with the learning environment.

This environment can be realized with a paper and pencil (no one is forbidden to write code on a piece of paper) but also through software in an integrated development environment that offers multiple opportunities for children to design the interface of their program and to write the code.

The choice of software is never simple: in order to give students rich and exciting experiences, it is necessary to find the right balance among the educational proposals, the age of the students and the possibilities (and boundaries) of the learning environment through which the experience of children (thoughts, beliefs and emotions) is filtered and given expression.

These factors can affect the perceptual and cognitive involvement of students and have an influence on their attention and motivation [5][8]. Therefore, it is important to learn more about the cognitive processes involved in the interaction with the coding environment, as well as how students' beliefs (expectations, skills and confidence of success) are modified by the experience of coding. This allows us to identify the aspects that promote the commitment of the students, and the factors that hinder the successful implementation of educational activities regarding coding.

As underlined in the Digital Agenda of the European Commission, to give children experiences of coding would not only promote success in computer science but would also increase creativity and teach collaboration [9].

Nevertheless, coding is still an optional subject in many Italian schools. It is difficult to organize activities on this topic due to problems of space, availability of computers and the organization of teaching schedules, as well as problems related to in-service teacher training.

However, numerous studies have shown that it is possible to meet this challenge in middle schools [10][11][12], although many of these educational activities about coding are delegated to after-school or summer workshops.

These initiatives are often developed by willing teachers or groups devoted to coding and, therefore, can be independent of the school curriculum itself.

We can see that there are still many issues that need to be addressed in Italy, especially now that a new school reform has been established which provides for the introduction of coding and computational thinking in primary and lower secondary schools [13].

In recent years, research has paid special attention to these issues, shifting focus from high school and university to primary and lower secondary schools (see section 2). However, current investigations focus mainly on issues related to the teaching and use of particular programming environments, although there is much to learn about cognitive processes and emotional aspects that can affect self-efficacy and learning engagement in future student experiences of coding.

The work presented in this article is an exploratory study that aims to investigate the themes of self-efficacy and learning engagement by students when involved in actual coding. In particular, the research questions which we will try to answer are: (a) How do the activities of coding affect students' beliefs about their self-efficacy? (b) How deep is the involvement level of cognition and perception of students during coding? (c) What are the main obstacles students can encounter while coding?

The exploratory study was carried out in a workshop on coding held in a first-level class of lower secondary education. It was presented as a question of relevant initiative on education/pedagogy in order to expand research studies of the sector, in addition to encouraging interested schools, teachers and students to participate in coding activities.

The aim is to provide results that serve as a starting point for further investigation of the teaching of coding in middle school, and to promote examples of educational activities to raise awareness of the subject and to exemplify the teaching of coding itself.

RELATED WORK

During the last thirty years, most research publications regarding CS focused on the teaching of programming at high school and university levels [14].

Over the past decade, however, there has been an increase in educational research in the field of coding within the primary and low secondary schools, largely through the efforts of the scientific community to redefine the concept of Computational Thinking [15], defined as *"[…] the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information processing agent*" [16] and the consequent redefinition of the digital skills necessary in the 21st century, which include *reading, 'riting,' 'rithmetic' and 'rithms'* (where' 'riting', 'rithmetic' and' 'rithms' stand for writing, arithmetic and algorithms) [14].

The research literature has focused mainly on the use of graphics, multimedia and blocks educational programming languages, which are based on the concept of "*a low floor* (easy to get started) and *a high ceiling* (opportunities to create increasingly complex projects over time)" [17]. Examples are applications such as 3D Alice, Scratch, Kodu, Logo and others, used mainly in primary and secondary levels, considered as introductory to the coding environment; and, in later levels of education, languages such as Java or C [18][19]. Often the use of these languages is added to activities such as programming and designing games [20] or in educational robotics activities [21][22].

In this context, many studies show high student engagement and document valuable learning outcomes which result from coding and computational thinking [23][18][24][25, made not only in mathematics and sciences [26][27] but also in teaching foreign languages [28], the visual arts [29] and fashion [30].

Other research has instead focused on innovative educational methods to teach coding, such as digital storytelling [31][32], collaborative learning [33] and problem-based learning [34]. However, only a few studies have focused on the teaching of coding to disabled students [35]. More studies are needed of gender differences [36][37] and the formalization of the evaluation of computational thinking [38]. According to [14] and [16], these research areas need more attention and "much remains to be done to help develop through more lucid theoretical and practical understanding of computational thinking in children" [14].

Our work is inspired by these references and other recent empirical research developments in this direction which are focused on the motivational and cognitive dimensions of coding and computational thinking [11][39][40]. For this reason we decided to investigate such aspects of the cognitive area as self-efficacy [6] and learning engagement [5], which can influence decisively the beliefs, perceptions, attitudes and motivation of children during their first experiences of coding. The aim is to acquire further information on those aspects that can help or hinder decisive learning and students' beliefs regarding coding, thereby contributing to a small but growing body of work on this issue within Italy and internationally.

CODING LABORATORY DESIGN

Scratch software, a project of the Lifelong Kindergarten Group of MIT Media Lab (https://scratch.mit.edu/about/), was used in establishing the coding laboratory.

This software features a logical approach to programming that is disconnected from any particular language, a wealth of media possibilities (including the adding of sound, text and images), high interaction and a high sensory and affective involvement, all of which have a notable impact on the cognitive involvement of students [8].

The learning activity which we studied took place in the province of Treviso and involved 20 students in the first level of a secondary school.

The coding activity was not characterized only by the use of Scratch, but accorded with the following four stages:

1. reading a story to the class (Aesop's fable entitled "The Fox and the Crow");
2. construction of a robot using Lego WeDo kit;
3. Scratch programming
    a. sketch interface (stage and sprites);
    b. development of the code associated with each element of the interface (sprites);
4. interaction between the program written in Scratch and robots built with Lego.

The first phase was crucial to involving students in the task and to contextualize the activity. Narrative was used to emotionally engage students and involve them through discussion of the story and setting out the goals to be achieved (stimulus questions and educational objectives are set out in section 4). After reading the story, the students were asked who were the protagonists and what actions were performed by each of them.

After defining the key players and the secondary ones, the class was divided into groups of three people; each group included pupils of the same sex, thereby creating three groups of only females and four groups of only males. Each group was given a Lego WeDo kit (LEGO education, URL: https://education.lego.com/en-us/lesi/elementary/lego-education-wedo) and they were told to build, with a maximum of 10 pieces, the fox (one of the protagonists in the story).

After this phase of construction, the Scratch program was introduced to the class. This software was previously installed in laptops delivered to the students (one laptop per group). After a brief introduction to the program, the students were asked to design the background of the story and insert into it, using sprites, the protagonists of the fable (the fox, the crow and the cheese).

After these early stages of construction and design, in which the students were left free to express their creativity, the development phase of the code began.

During this phase, the pupils were led to implement the functions associated with each actor of the program. This phase took place interactively according to the following steps: a brief explanation of the Scratch commands by the researcher, testing of them by the students and finally demonstration by the researcher of a possible application of the commands through the use of a digital whiteboard.

In the last activity phase, after the implementation of the program associated with each actor in the story, the robot built by each group was connected to a gyroscope, and the gyroscope to the computer. In this way the angle of movement of the fox was transmitted to the Scratch program. After a small code modification, the students managed to move the foxes in their programs, visualized in their monitors, according to the angle at which the fox built with Lego was bent.

RESEARCH METHODOLOGY

As noted in the introduction, this paper aims to answer three basic questions:

- how the activity of coding effects students' self-efficacy beliefs?
- How deep are students' levels of cognitive involvement and perception during the activity of coding?
- What are the principal obstacles that students might encounter when coding?

To answer these questions, a workshop on coding was organised (see section 3), which was preceded and followed by the administration of 1) a pre-test questionnaire that required general information and estimates of self-efficacy; 2) a post-test questionnaire on self-efficacy and cognitive involvement.

The first part of the pre-test questionnaire sought information on gender, age and the use of computers. Specifically, it asked the students if they were able use a computer at home, do they ever browse the internet, what software do they use for fun, and do they know how to download programs from the net.

The second part of the pre-test and post-test questionnaire asked students to complete the New General Self-Efficacy Scale (NGSE), which aims to explain the variance in motivation and performance by measuring the degree of self-efficacy of the students [7]. Bandura [6] defines self-efficacy as "*people's beliefs about their capabilities to produce designated levels of performance that exercise influence over events that affect their lives" and he states that these beliefs "determine how people feel, think, motivate themselves and behave*". Many studies have already shown that coding is a complex and difficult task for students of lower secondary school, as it requires skills inherent in the resolution of problems, such as abstraction, logic and mathematics [41].

There are two reasons why it is important to obtain data on the degree of self-efficacy. First, studies undertaken outside Italy have shown that students' prior experiences of coding may influence their subsequent performance [10][12]. Secondly, we know of no studies in Italy that have investigated this issue, as the teaching of coding is a relatively new field in Italy and is only taught in a few schools.

The questionnaire was based on 4-items (each measured on the five-point Likert scale), which were scored from 1 (completely disagree) to 5 (completely in agreement).

It investigated four relevant matters:

- the level of self-efficacy in learning activities and understanding of coding;
- the expectation of performing well in one's own activities;
- the security of reaching a successful conclusion in one's own activities;
- the self-confidence of obtaining excellent results from the activity of coding.

The second questionnaire, administered after the workshop on coding, is based on (post-test) self-efficacy measurement and the User Engagement Scale [8] which measures the degree of students' engagement in the activity of learning coding.

In the version originally published, the User Engagement Scale is a questionnaire consisting of many questions, but with regard to the composition of our sample and the context in which the activity took place, it was decided to limit the items to eight.

The reasons for this choice are related to the young age of the respondents and the fact that we would already have given them a questionnaire before starting the workshop and would be submitting them to a second survey, after three hours of classroom activities.

Due to these consideration, we decided to limit our investigation to the following aspects which affect cognitive involvement of students:

- *Focused attention*: the degree of concentration perceived by students while they were working with Scratch. Specifically, students were asked to assess their attention during the coding activities, with reference to the extent of their ignorance of other stimuli, such as time and space [42]. (Question 1: "I was so involved in my work programming in Scratch that I ignored everything around me"; Question 2: "I was so involved in my task of programming in Scratch that I lost track of time.").
- *Novelty*: defined as a state of excitement, curiosity, joy and alarm [43], which can occur when people use a program that they find new, unfamiliar, unexpected or surprising [44]. In this investigation, the appearance of novelty was considered a key element, since it was assumed that the software used to program, Scratch (see paragraph 3), would be a new experience for the students. (Question 3: "Coding activity incited my curiosity; Question 4: "I discovered by doing this activity that I like to program.")
- *Involvement*: defined as "a need-based cognitive (or belief) state of psychological identification with some object that is based on an individual's salient needs and perception that the object will satisfy those needs" [45] (Question 5: "Programming was very funny.")
- *Aesthetics*: an investigation of the aesthetic impressions regarding the interface of a program, such as the screen layout and images (Question 6: "I liked the graphics and images used in Scratch.")
- *Perceived usability*: a factor that explicitly investigates perceptions of difficulties experienced by students during their coding tasks. This is important in assessing the students' perceived effort while they are performing their assigned exercises. (Question 7: "I found it difficult to use Scratch for coding."; Question 8: "I felt discouraged while I was using Scratch"; Question 9: I could not do some of the things I had to do with this software.")

Sixteen questions were included in the first version of the second questionnaire; these were divided between those we considered fundamental and those we considered less important; next, we discussed each item and decided on the final version.

The questionnaire was then assessed by two researchers: a developmental psychologist and an expert in educational technologies. Both gave positive opinions on the questionnaire with a few suggested revisions.

The developmental psychologist considered the questions readable and understandable for 11-year-old students but suggested dividing the item on attention into two questions: one on time and the other on space. The educational technology expert recommended extrapolating from the various items in the Scratch program a specific question concerning the graphical interface of the program.

The nine answers to the second questionnaire (in line with the pattern used for the first one) were based on the five steps of the Likert scale; they ranged along an axis from 1 (completely disagree) to 5 (completely agree).

To further verify the students' experience of coding, we decided to analyze videos of the operations the students performed during the workshop. The purpose was to examine what students learned in practice and what their difficulties were. The criteria used to evaluate the videos were based on the educational goals of the laboratory, as listed below:

- *Define the essential elements of a problem.* To know how to identify the main components of a problem is extremely important in computer sciences (e.g., diagram of the classes of a program; the conceptual schema of a database) and it is often the most critical element in the initial phase of coding. *Questions posed to the students*: What are the main actors of the story? *Activities with Scratch*: The main objects of the story were represented by sprites, while the secondary elements, which it was not necessary to animate, were placed on the stage.
- *Identify the actions associated with each object of the problem and when it is necessary to activate them.* After identifying the main objects of a problem, one needs to define their behavior; that is, what they must do and when they have to perform particular actions (e.g., object methods, event management interface). This step is crucial before moving to the implementation of behavior and the methods of effectuating certain actions. *Questions posed to the students*: What actions are performed by the main actors of the story? Which character moves first? When should you move the cheese? What should the fox do as the cheese falls? *Activities with Scratch*: *Control* commands (*when [] key pressed*, *broadcast []* and *when I receive []*).

- *Describe the termination condition of a cycle (construct iteration)*. In programming it is often necessary to use an iterative structure, especially if you use languages that are part of the procedural paradigm. In the laboratory one focuses on the structure of indefinite iteration, used when you do not know in advance how many times to repeat a block of instructions. The main problem when employing this construct is to understand what condition should be specified to end the iteration. *Questions posed to students*: When to stop the cheese? How far must it come down? *Activities with Scratch*: *Control* commands (*repeat until []*), *Sensing* commands (*touching [] ?*).
- *Using variables*. The concept of a variable is important in coding. In particular, we need to understand how to read data from one variable and assign it a new value. The most difficult thing to understand is how you initialize a variable (Which value contains a variable you have just created?) and how you distinguish between the name of a variable and its value. *Questions posed to the students*: How do you move the cheese down? Specify what value to assign to variable y? How do you bring the cheese to the starting position each time the program begins? *Activities with Scratch*: *Motion* commands (*x position*, *y position*, *change x by []*, *change y by []*, *set x to []*, *set y to []*).
- *Describe the condition required to select a block of code to be executed (the construct of selection)*. As with the iterative structure, the structure of selection is important in the activity of coding. During the workshop activities, emphasis was placed on the unary selection structure, employing examples of cascading constructs (when two or more identical constructs must be performed in succession). In this case it is important to specify the condition for passing control of the selection and execution of the required block of code. *Questions posed to students*: When to move the fox right? When to the left? In consequence of which sensor value, should it "get up"? *Activities with Scratch*: *Control* commands (*if []*), *Sensing* commands (*[tilt] sensor value*), *Operators* commands (*[] = []*).
- *Distinguish between the input and output of a program*. One of the first matters to understand in the development of a program is how to take "input" data and provide "output" as a result. In addition to the difficulty of understanding what commands to use to carry out these tasks, the students often fail to understand which is the input and which is the output. Sometimes this problem is associated with the difficulty of distinguishing the input / outputs that are connected to the computer. This is one of the reasons why Lego robots are used (see section 3): to emphasize, through sensors and actuators, the flow of data going to the

computer and coming out from it. *Questions posed to the students*: How to make a crow sound? How to clear the cheese eaten by the fox? How to move the cheese from the crow to the fox? How to show that the fox is happy to have taken the cheese? How to move the fox according to the motion sensor "get up"? *Activities with Scratch*: *Looks* commands (*say [] for [] secs*, *hide*, *show*), *Motion* commands (*change x by []*, *change y by []*), *Sensing* commands (*[tilt] sensor value*).

RESEARCH RESULTS

Sample

The sample consisted of 20 students from a secondary school class (first level), who participated in the pilot study during school hours. Although the size of the sample is small, it is sufficiently significant of the secondary school population. The demographic composition is 11 males and 9 females aged between 11 and 12 years, including 4 foreign males and 2 foreign females. Among the students there is a dyslexic child.

Out of 20 students, only one has not the opportunity to use the computer at home and to surf on Internet. Most of the students stated they use a word processor, Google search engine and YouTube to watch movies and their favourite cartoons. Only one student stated to have already used Scratch.

Self-Efficacy Questionnaire Results

Before analyzing the results, the reliability of the questionnaire was estimated by calculating Cronbach's alpha coefficient. The calculation of this index had a good result ($\alpha = 0.792$), and this value was also confirmed by items statistics (Table 1). In fact, when trying to remove iteratively an item from the questionnaire, the value of α remains under 0.775. Further, the corrected item-total correlation is higher than 0.4 for each item, demonstrating good internal consistency of the questions.

Analysis pre- and post-test regarding self-efficacy. To evaluate whether the learning activity produced an improvement in the sense of self-efficacy, a paired samples test was performed (Table 2) which revealed that the change (Pre: M = 3.925, SD = 0.597, range 3-4.5; Post: M = 4.538, SD = 0.521, range 3.25-4.75) was statistically significant ($t(19) = -3.862$, $p < 0.01$).

Before calculating the Student t test, a normality test was conducted to make sure data were distributed in a Gaussian way. In particular, since our sample was composed of fewer than 50 people, the Shapiro-Wilk test was chosen, getting sig. value equal to 0.662, confirming the normal distribution of data (it is possible to confirm this result also observing Figure 1).

Table 1
Items Statistics

	Corrected Item-Total Correlation	**Alpha if Item Deleted**
Q2.1	0.529	0.775
Q2.2	0.733	0.673
Q2.3	0.532	0.774
Q2.4	0.626	0.730

Table 1
Items Statistics

	t	**df**	**Sig. (2-tailed)**
Pre - Post	-3.862	19	0.001

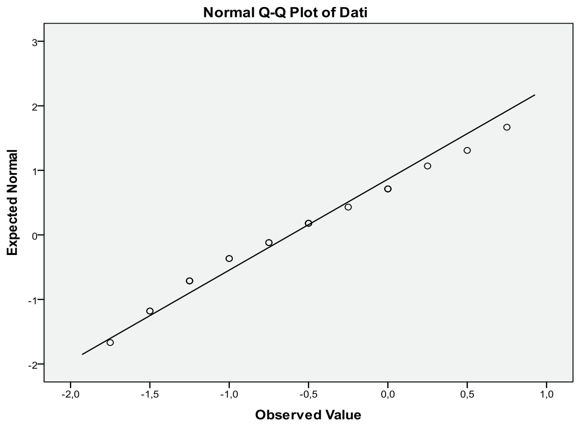

Figure 1. Normal Q-Q Plot.

User Engagement Questionnaire Results

The questionnaire analysis put in evidence a low affinity between questions 4.1 (“I was so involved in my programming work with Scratch that I ignored everything around me”) and 4.2 (“I was so involved in my programming task with Scratch that I lost track of time”), as highlighted by the corrected item-total correlation and by the inter-item correlation matrix (value 0.281). In addition, the same questions were unreliable to measure “Focused attention” ($\alpha = 0.417$).

Similar results were detected regarding questions 4.3 (“programming activity stimulated my curiosity”) and 4.4 (“I have discovered by doing this activity that I like to program”) used to detect “Novelty”. In fact, these items got a Cronbach’s alpha coefficient equal to 0.417 and from the inter-item correlation matrix a value of 0.264.

The mean value of the responses to questions 4.1, 4.2, 4.3, 4.4 ($M > 4.25$, in a range from 1 to 5) suggests a strong involvement of the students in the coding activity. This value was confirmed also by question 4.5 (“This programming experience was very enjoyable”), which obtains an average value equal to 4.6.

A deeper analysis of the questionnaire revealed that considering together the items from 4.1 to 4.5, it is possible to obtain a significant index of reliability ($\alpha = 0.757$) and a good affinity between the questions (Tables 3 and 4).

Table 3

Inter-Item Correlation Matrix

	Q4.1	Q4.2	Q4.3	Q4.4	Q4.5
Q4.1	1.000	0.281	0.333	0.175	0.706
Q4.2	0.281	1.000	0.562	0.365	0.434
Q4.3	0.333	0.562	1.000	0.264	0.542
Q4.4	0.175	0.365	0.264	1.000	0.382
Q4.5	0.706	0.434	0.542	0.382	1.000

Table 4
Items Statistics

	Corrected Item-Total Correlation	Alpha if Item Deleted
Q4.1	0480	0.729
Q4.2	0.545	0.732
Q4.3	0.600	0.693
Q4.4	0.390	0.756
Q4.5	0.712	0.654

Items used to measure different factors: Focused Attention, Novelty and Involvement seem to converge in the measurement of a single factor, which it is possible to define with the term "Novelty". In fact, since the experience carried out by the students was something new, achieved through the presence of new teachers, it is likely that the factor of "Novelty" had a strong impact on "Focused Attention" and "Felt Involvement", as demonstrated by other studies [8]. This led us not to distinguish the different factors.

According to the questions analyzed above, item 4.6 ("I liked the graphics and images used in Scratch") concerning "Aesthetics" had positive answers (M = 4.45). This demonstrates that the software chosen for the task had a positive impact on the cognitive involvement of the students.

Finally, the analysis of items 4.7 ("I found Scratch difficult to use for programming"), 4.8 ("I felt discouraged while I was using Scratch") and 4.9 ("I could not to do some of the things that I had to do with this software"), used to detect the usability perceived by students, has a Cronbach's alpha coefficient of 0.668. As can be seen from Table 5, the reliability index has been improved by deleting item 4.9, thereby obtaining a Cronbach's alpha coefficient equal to 0.684.

Table 5
Items Statistics

	Corrected Item-Total Correlation	Alpha if Item Deleted
Q4.7	0.508	0.541
Q4.8	0.539	0.500
Q4.9	0.409	0.684

Also in this case, the choice of deleting item 4.9 is connected to the activities undertaken. Since the researcher guided the students in using

Scratch, almost everyone reached the same level, completing all assigned tasks. Moreover, working in groups, the students were able to find support and implement all the tasks required. Given this background, while the items 4.7 and 4.8 are questions of self-evaluation about the activity, item 4.9 may have appeared unclear to the students. In fact, to answer this item it would have been necessary to have a deeper knowledge of coding in order know what it was possible to do or not to do with Scratch software.

The average of the answers to questions 4.7, 4.8 and 4.9 were respectively 2.75, 1.7 and 2.25, in a range from 1 (completely disagree) to 5 (completely agree), which demonstrates that the students have had few difficulties in using Scratch, probably as a result of the software graphics (see answer 4.6).

Analysis of Results of Video and Programs

To analyze the students' work during the lab session, the following information was taken into account: the answers given by the students to the stimulus questions (see section 4); the videos made through a video recording software containing their interactions with Scratch; the programs implemented by the students.

The analysis of the data, conducted by researchers after the lab session, was intended to evaluate the achievement of the objectives described in section 4, detect the difficulties faced by the students during coding, and collect information regarding students' involvement and their reactions to tackling obstacles. The results of the analysis follow:

- *Define the essential elements of a problem. Questions posed to the students*: What are the main actors of the story? *Data analysis*: all students immediately identified the fox and the crow as the mainstays of the story. Only four out of 20 students, however, noted that the cheese was an important entity of the problem – this too was represented by a sprite and programmed.
- *Identify the actions associated with each entity of the story and, when necessary, activate them. Questions posed to the students*: What actions are performed by the main actors of the story? Which character moves first? When do you have to move the cheese? What should the fox do while the cheese falls? *Data analysis*: all of the students realized that the crow was the first character to be animated, but most of them hesitated when deciding when to drop the cheese and what the fox should do as the cheese descended. The researcher then intervened to explain to the

students how to send a message from one sprite to another (the cheese begins to fall when the raven sends a message). After this intervention, all of the children were able to synchronize the various sprites. Only one group (three boys) encountered problems at this point, the wrong use of commands: *broadcast [ ]* and when *I receive [ ]*. The same group also wrongly used the command *wait until [ ]* to synchronize the fox.

- *Describe the termination condition of a cycle (repetition construct). Questions posed to students*: When does the cheese stop? How far must it come down? *Data analysis*: all students successfully used the construct iteration to define problems that didn't have a termination condition.
- *Using variables. Questions posed to the students*: How to move the cheese down? What is the specific value for the variable y? How to bring the cheese to the starting position each time the program is run? *Data analysis*: the majority of students had problems in understanding the meaning of the variables *x position* and *y position*, writing the corresponding values randomly in commands *set x to [ ]* and *set y to [ ]*. In most cases the researcher intervened to resolve the issue. Problems with the variables occurred during the integration of the Lego robots with Scratch: one group (3 students) did not understand the difference between the variable representing the speed of the fox and the variable containing the value of the sensor movement.
- *Describe the condition required to select the block of code to be executed (selection construct). Questions posed to students*: When to move the fox to the right? When to the left? In consequence of which sensor value should the fox "get up"? *Data analysis*: all students successfully used the construct of selection, defining the condition without problems.
- *Distinguish between the input and output of a program. Questions posed to the students*: How to make a sound like a crow? How to clear the cheese eaten by the fox? How to move the cheese from the crow to the fox? How to show that the fox is happy to have taken the cheese? How to move the fox according to the motion sensor "get up"? *Data analysis*: almost all the students were able to distinguish between the input and the output of the program, although some boys were hesitant to consider the movement of the sprite as an output, expecting, instead, to see the movement of the fox which they had developed. For many pupils, the output was considered as something that comes out from the computer, not as the result of processing.

From analysis of the data collected by video it is interesting to note that during the coding stage only one group of the seven respected to the letter the directives of the researcher, while the other six groups tried to customize their code (different names for sprites, different values for the variables). Of these six groups, only two made customizations that improved the quality of the program, the other four encountered some errors (name of the sprite too long or erroneous, values outside the scale for movements, confusion between the identification sensor and speed of movement).

Although there were several mistakes made by students during the writing of the code, it is interesting to note that the groups were able to self-correct in 30% of cases, often without finding the optimal solution but still reaching the desired objective. The search for a solution to a problem allowed the students to deepen their study of the tools offered by Scratch and make a product different from that proposed by the researcher although still acceptable and functional.

Albeit late, and close to the sound of the bell that signaled the end of the class, all of the groups wanted to modify their programs to make sure that the data read from an external sensor, so interfacing with their previously built Lego artifact. In addition, at the end of the lesson, most of the children were eager to continue the development of the program to add new features (their motivation led them to work for three hours in a row without leaving the classroom).

LIMITATIONS

This study has several limitations and constraints which are often encountered during research in classroom settings. First, it was carried out on a class selected at random, according to the hourly availability of the teachers of the institution. As a result, it was not possible to make comparisons with other classes in the same school. In addition, the study was performed during the last weeks of the school year. Consequently, the demands of the experiment had to co-exist with other school activities that required attention and commitment. Also, the students were aware they were participating in a three-hour experiment that would not influence their school assessment. Despite these limitations, all of the students completed the questionnaire, without any reduction in the sample size.

The enthusiasm shown by the students could be due in part to their belief that they were involved in “an older students’ activity”. Additionally, the novelty factor may have biased the results on user engagement elicited in the questionnaire (see section 5, paragraph C).

Employing a variety of investigation strategies would assist the interviews at individual and group levels and also improve the grids of participant observation. These elements would strengthen the study and will be considered for our future research.

It would also have been rewarding to be able to return a week later in order to verify how much knowledge and enthusiasm the students retained, and to initiate a new proposal for coding at an advanced level.

A final consideration (connected with paragraph 2) is that it would have been interesting to investigate the differences between males and females, as in this sample the numbers are balanced. Nonetheless, we believe it is necessary to begin with an assessment of the results of the whole class and in future to have regard to gender differences in this sample, if they are significant.

DISCUSSION AND CONCLUSION

The questionnaire on self-efficacy demonstrates that the activity of coding has resulted in an improvement, though small (0.6 points), in the students' sense of self-efficacy, which in any case was already quite high. The pre-test, in fact, reported an average of 3.925, with a range of values from 3 to 4.5 points (minimum 1, maximum 5). The high level of self-efficacy is supported by the analysis of the video: most of the groups tried to customize their program, a sign of their self-confidence in carrying out various tasks. In light of these results it is possible to respond positively to the first research question: Does the activity of coding affect students' beliefs in their self-efficacy?

This result is attributable to the coding software used (Scratch) and the organization of the activity, which allowed the students a new and engaging learning experience. As shown by the analysis of the questionnaire on user engagement, this new element had a positive impact on the attention and involvement of the students. The children themselves reported that the programming tool, Scratch, was aesthetically eye-catching and easy to use. Analysis of the video supports this claim, as it shows many groups trying to self-correct their errors, a symptom of good software usability. The cognitive and perceptive involvement of the students is confirmed both by the results of the Scale User Engagement questionnaire and by analysis of videos, which revealed, in particular, the students' desire to add new features to the program and to repeat their experience of coding.

Nevertheless, although, the coding activity achieved good results, from video analysis we have found various difficulties, as already shown in other studies [46][47]. These are mainly related to certain abstract concepts, such as the use of variables, the identification of the essential elements of a problem, the sequencing of events. These all require time and practice to be fully assimilated.

Notes

The authors received no financial support for the research and / or authorship of this article.

For reasons regarding the national assessment of Italian university research, the authors must declare which sections each has written, in spite of the fact that the entire article is the result of continuous and intensive collaboration. Therefore, we declare that I. Introduction, II. Related Work, IV. Research Methodology, VI. Limitations, VII. Discussion and Conclusion are by Monica Banzato; III. Coding Laboratory Design, V. Research Results are by Paolo Tosato.

References

[1] D. H. Jonassen. *Computers as mindtools for schools: Engaging critical thinking.* Prentice Hall, 2000.

[2] G. Polya. *Mathematical discovery: On understanding, learning, and teaching problem solving*. 1981.

[3] J E. Davidson, R. Deuser and R. J. Sternberg. *The role of metacognition in problem solving*. 1994.

[4] R. E. Mayer. *Thinking, problem solving, cognition*. WH Freeman/Times Books/Henry Holt & Co, 1992.

[5] P. R. Pintrich and D. H. Schunk. *Motivation in education: Theory, research, and applications*. Prentice Hall, 2002.

[6] A. Bandura. *Self-efficacy: The exercise of control*. New York: Freeman, 1997.

[7] G. Chen, M. G. Stanley and E. Dov. “Validation of a new general self-efficacy scale”. *Organizational research methods* 4.1 (2001): 62-83.

[8] H. L. O’Brien and G. T. Elaine. “The development and evaluation of a survey to measure user engagement”. *Journal of the American Society for Information Science and Technology* 61.1 (2010): 50-69.

[9] European Commission (2015, July 20). *Digital Agenda for Europe, Coding - the 21st century skill* [online]. Available: http://ec.europa.eu/digital-agenda/en/coding-21st-century-skill.

[10] A. Arpaci-dusseau, O. Astrachan, D. Barnett et al. “Computer Science Principles: Analysis of a Proposed Advanced Placement Course”. *Proceedings of SIGCSE ’13* (2013), 251-256.

[11] P. S. Buffum, A. G. Martinez-Arocho, M. H. Frankosky et al. “CS principles goes to middle school: learning how to teach Big Data”. Proceedings of the *45th ACM technical symposium on Computer science education*, March 2014, pp. 151-156. ACM.

[12] M. Guzdial, B. J. Ericson, T. McKlin et al. “A Statewide Survey on Computing Education Pathways and Influences: Factors in Broadening Participation in Computing”. Proceedings of the *ninth annual international conference on International computing research* (ICER ‘12) (2012), 143–150.

[13] MIUR (2014, July 25). *La Buona Scuola. Facciamo crescere il Paese* [online]. Available: https://labuonascuola.gov.it/documenti/La%20Buona%20Scuola.pdf?v=d0f805a

[14] S. Grover and R. Pea. *Computational Thinking in K–12 A Review of the State of the Field. Educational Researcher*, 2013, 42(1), 38-43.

[15] J. M. Wing. *Computational thinking. Communications of the ACM*, 2006, 49(3), 33-35.

[16] J. M. Wing. “Computational thinking”. In *VL/HCC*. 2011, p. 3.

[17] M. Resnick, J. Maloney, A. Monroy-Hernández et al. “Scratch: programming for all”. *Communications of the ACM*, 2009, 52(11), 60-67.

[18] U. Wolz, H. H. Leitner, D. J. Malan et al. “Starting with scratch in CS 1”. In *ACM SIGCSE Bulletin.* 2009, Vol. 41, No. 1, pp. 2-3, ACM.

[19] M. Armoni, O. Meerbaum-Salant and M. Ben-Ari. “From scratch to “real” programming”. *ACM Transactions on Computing Education (TOCE)*, 2015, 14(4), 25.

[20] A. R. Basawapatna, K. H. Koh and A. Repenning. “Using scalable game design to teach computer science from middle school to graduate school”. In Proceedings of the *fifteenth annual conference on Innovation and technology in computer science education*, 2010, pp. 224-228, ACM.

[21] D. Alimisis and C. Kynigos. “Constructionism and Robotics in education”. *Teacher Education on Robotic-Enhanced Constructivist Pedagogical Methods*. 2009, pp. 11-26.

[22] K. J. O.’Hara and J. S. Kay. “Investigating open source software and educational robotics”, published in *Journal of Computing Sciences in Colleges*, Volume 18, Issue 3, February 2003, pp. 8-16, Consortium for Computing Sciences in Colleges, USA, ISSN:1937-4771 EISSN:1937-4763.

[23] U. Wolz, J. Maloney and S. M. Pulimood. “‘Scratch’ your way to introductory CS”. *SIGCSE Bull*. 2008, 40, 1, pp. 298-299.

[24] J. Maloney, K. Peppler, Y. Kafai et al. “Programming by choice: Urban youth learning programming with Scratch”. Proceedings of *SIGCSE conference*, 2008.

[25] L. Werner, S. Campe and J. Denner. “Children learning computer science concepts via Alice game-programming.” Proceedings of *the 43rd ACM technical symposium on Computer Science Education.* ACM, 2012.

[26] O. Meerbaum-Salant, M. Armoni and M. Ben-Ar. *Learning computer science concepts with scratch. Computer Science Education*, 23(3), 2013, pp. 239-264.

[27] K. T. Stolee and T. Fristoe. “Expressing computer science concepts through Kodu game lab”. In Proceedings of the *42nd ACM technical symposium on Computer science education* (2011, pp. 99-104). ACM.

[28] J. Solomon (2004, October 12), *Programming as a Second Language, in Learning & Leading with Technology* [online]. Available: https://people.csail.mit.edu/jsolomon/assets/secondlanguage.pdf

[29] S. H. Rodger, J. Hayes, G. Lezin et al. "Engaging middle school teachers and students with alice in a diverse set of subjects". ACM *SIGCSE Bulletin*. Vol. 41. No. 1. ACM, 2009.

[30] W. W. Y. Lau, G. Ngai, S. C. F. Chan et al. "Learning programming through fashion and design: a pilot summer course in wearable computing for middle school students". ACM *SIGCSE Bulletin*. Vol. 41. No. 1. ACM, 2009

[31] C. Kelleher and R. Pausch. "Using storytelling to motivate programming". *Communications of the ACM*, 50, 7, 2007, pp. 59-64.

[32] B. Quinn and Y. B. Kafai. "The writers' workshop for youth programmers: digital storytelling with scratch in middle school classrooms". Proceedings of *the 43rd ACM technical symposium on Computer Science Education*. ACM, 2012.

[33] Y. B. Kafai, D. A. Fields and Q. Burke. "Collaborative agency in youth online creative production in Scratch". In Proceedings of *the 19th International Conference on Computers in Education* (Chiang Mai, Thailand, 2011 Nov 28 – Dec 2).

[34] J. Arlegui, A. Pina and M. Moro. "A PBL approach using virtual and real robots (with BYOB and LEGO NXT) to teaching learning key competences and standard curricula in primary level". Proceedings of the *First International Conference on Technological Ecosystem for Enhancing Multiculturality*. ACM, 2013.

[35] J. C. Adams. "Scratching middle schoolers' creative itch". In Proceedings of *the 41st ACM technical symposium on Computer science education.* 2010, pp. 356-360. ACM.

[36] J. Denner, L. Werner, S. Bean at al. "The girls creating games program: Strategies for engaging middleschool girls in information technology". *Frontiers: A Journal of Women Studies*, 2005, 26, 1, pp. 90-98.

[37] J. Denner, L. Werner and E. Ortiz. "Computer Games Created by Middle School Girls: Can They Be Used to Measure Understanding of Computer Science Concepts?". *Computers & Education*, 2012, 58(1), pp. 240-249.

[38] L. Werner, J. Denner and S. Campe. "The fairy performance assessment: Measuring computational thinking in middle school". Proceedings of *the 43rd ACM technical symposium on Computer Science Education*. 2012, ACM: Raleigh, North Carolina, USA. pp. 215-220.

[39] K. Qiu, L. Buechley, E. Baafi et al. "A curriculum for teaching computer science through computational textiles". Proceedings of *the 12th International Conference on Interaction Design and Children*, pp. 20-27, June 24-27, 2013, New York, New York.

[40] C. Chang. "Effects of Using Alice and Scratch in an Introductory Programming Course for Corrective Instruction". *Journal of Educational Computing Research*. 51.2 (2014): pp. 185-204.

[41] M. J. Saeli, W. Perrent and B. Jochens Zwaneveld. "Teaching programming in secondary school: A pedagogical content knowledge perspective". *Informatics in education. Informatics in Education*, 2011, 10(1): 73-88.

[42] M. W. Matlin. *Cognition (3rd ed.)*. Orlando, FL: Harcourt Brace, 1994.

[43] A. Aboulafia and L. J. Bannon. “Understanding affect in design: An outline conceptual framework”. *Theoretical Issues in Ergonomic Science*, 2004, 5(1), pp. 4–15.

[44] M. Huang. “Designing website attributes to induce experiential encounters”. *Computers in Human Behavior*, 2003, 19, pp. 425–442.

[45] L. A. Kappelman. “Measuring user involvement: A diffusion of innovation perspective”. *Database Advances*, 1995, 26(2/3), pp. 65–86.

[46] M. Iain and R. Glenn. “Difficulties in Learning and Teaching Programming—Views of Students and Tutors”, in *Education and Information Technologies*, March 2002, Volume 7, Issue 1, pp 55-66.

[47] A. Robins, J. Rountree and N. Rountree. “Learning and Teaching Programming: A Review and Discussion”, in *Computer Science Education*, Volume 13, Issue 2, 2003, pp. 137-172, DOI: 10.1076/csed.13.2.137.14200.